

11.2　Alpha滤镜

Alpha滤镜能实现针对图片文字元素的“透明”效果，这种透明效果是通过“把一个目标元素和背景混合”来实现的，混合程度可以由用户指定数值来控制。通过指定坐标，可以指定点、线和面的透明度。如果将Alpha滤镜与网页脚本语言结合，并适当地设置其参数，就能使图像显示淡入淡出的效果。

Alpha滤镜的语法格式如下：

```
    {filter : Alpha ( enabled=bEnabled, style=iStyle, opacity=iOpacity,
finishOpacity=iFinishOpacity,
                 startx=iPercent, starty=iPercent, finishx=iPercent,
finishy=iPercent ) }
```

各参数如表11-2所示。

表11-2 Alpha滤镜参数

参数	说明
enabled	设置滤镜是否激活
style	设置透明渐变的样式，也就是渐变显示的形状，取值为0~3。0表示无渐变，1表示线形渐变，2表示圆形渐变，3表示矩形渐变
opacity	设置透明度，值范围是0~100。0表示完全透明，100表示完全不透明
finishOpacity	设置结束时透明度，值范围也是0~100。
startx	设置透明渐变开始点的水平坐标（即x坐标）
starty	设置透明渐变开始点的垂直坐标（即y坐标）
finishx	设置透明渐变结束点的水平坐标
finishy	设置透明渐变结束点的垂直坐标

【例11.1】实例文件：ch11\11.1.html

```
    <html>
    <head>
        <title>Alpha滤镜</title>
    </head>
    <body>
          原始图<img src="baimd.jpg" style="width:200px;height:120px;">
            style=0<img src="baimd.jpg" style="width:200px;height:120px;filter
: Alpha(opacity=60 , style=0)" >
            style=2<img src="baimd.jpg" style="width:200px;height:120px;filter
: Alpha(opacity=60 , style=2)" >
           style=3 <img src="baimd.jpg" style="width:200px;height:120px;filter
: Alpha(opacity=60 , style=3)" >
      </body>
    </html>
```

在IE9.0中浏览效果如图11-1所示，可以看到显示了4张图片，其透明度依次减弱。

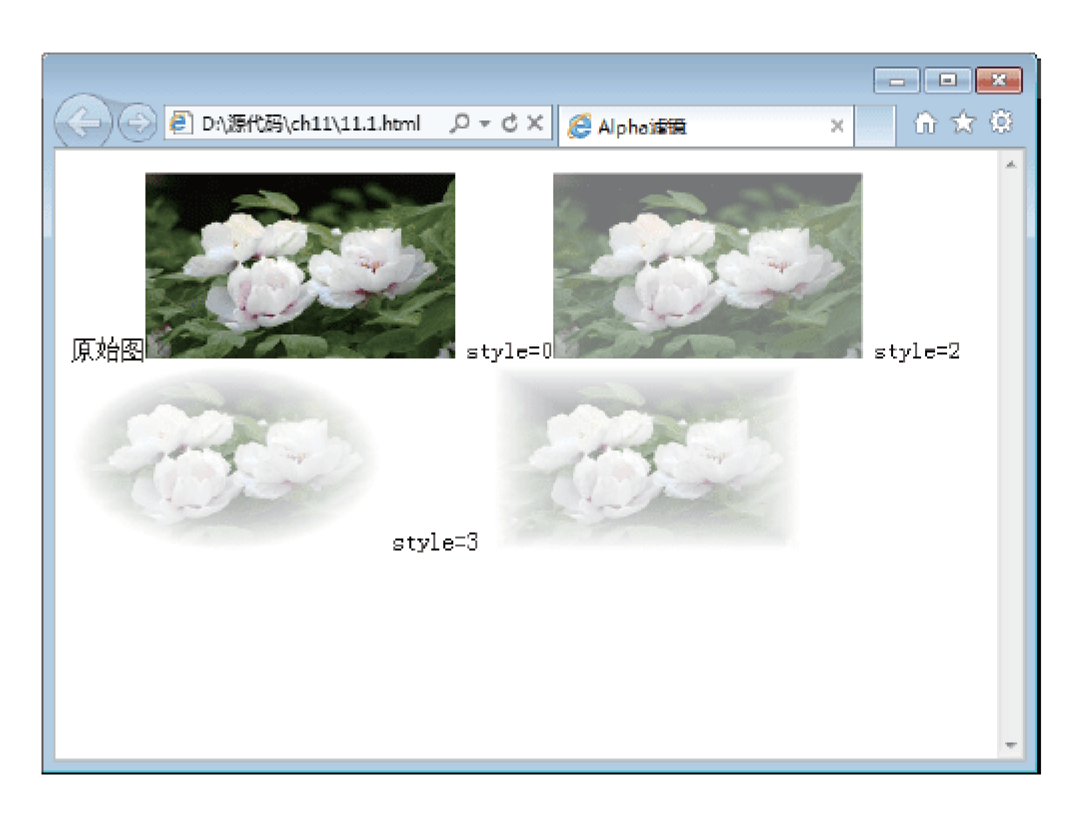


图11-1 实现图片滤镜效果

在使用Alpha滤镜时要注意以下两点：

(1) 由于Alpha滤镜使当前元素部分透明，该元素下层的内容的颜色对整个效果起着重要作用，因此颜色的合理搭配相当重要。

(2) 透明度的大小要根据具体情况仔细调整，取一个最佳值。

Alpha滤镜不但能应用于图片，还可以应用于文字透明特效。

【例11.2】实例文件：ch11\11.2.html

```
<html>
<head>
    <title>Alpha滤镜</title>
    <style type="text/css">
    <!--
      p{
        color:yellow;
        font-weight:bolder;
        font-size:25pt;
        width:100%
      }
    -->
    </style>
</head>
<body style="background-color:Black">
    <div >
      <p>Alpha滤镜</p>
      <p style="filter:alpha(opacity=60 , style=1)">透明效果</p>
      <p style="filter:alpha(opacity=60 , style=2)">透明效果</p>
      <p style="filter:alpha(opacity=60 , style=3)">透明效果</p>
    </div>
  </body>
</html>
```

在IE9.0中浏览效果如图11-2所示，可以看到显出现了4个段落，其透明度依次减弱。



图11-2　文字透明效果



11.3 BlendTrans滤镜

BlendTrans滤镜是一种高级滤镜，如果要实现效果，需要结合JavaScript。该滤镜可以实现HTML对象的渐隐渐现效果。

BlendTrans滤镜语法格式如下：

```
{ filter : BlendTrans ( enabled=bEnabled , duration=fDuration ) }
```

上述代码中，enabled表示是否激活滤镜；duration表示整个转换过程所需要的时间，单位为秒。

【例11.3】实例文件：ch11\11.3.html

```
<html >
<head>
    <title>BlendTrans滤镜</title>
    <style type="text/css">
    <!--
      .blendtrans { filter:blendTrans(Duration=3)}
    -->
    </style>
</head>
<body onload="playImg()">
  <img src="baimd.jpg" class="blendtrans" id="imgpic"  style="width:300px;
height:280px;" />
  <script language="JavaScript">
  <!--
    // 声明数组，数组元素的个数就是图片的个数，然后给数组元素赋值，值为图片路径
    ImgNum=new ImgArray(2);
    ImgNum[0]="baimd.jpg";
```

```
    ImgNum[1]="baihua.jpg";
    // 获取数组记录数
    function ImgArray(len)
    {
      this.length=len;
    }
    var i=1;
    //转换过程
    function playImg(){
      if (i==1){
        i=0 ;
      }
      else{
        i++;
      }
      imgpic.filters[0].apply();
      imgpic.src=ImgNum[i];
      imgpic.filters[0].play();
      // 设置演示时间，这里是以毫秒为单位的，4000则表示延迟秒
      // 滤镜中设置的转换时间值，这样当转换结束后还停留一段时间
      timeout=setTimeout('playImg()',4000);
    }
    -->
    </script>
</body>
</html>
```

代码中，对HTML元素img中应用了BlendTrans滤镜，然后使用JavaScript脚本语言来定义转换过程。对于JavaScript代码，要声明用来存储图片数组，并指定图片所在路径。然后，再获取数组长度，用于转换过程中循环读取图片数量。接着定义转换过程playimg，该过程实现了两幅图片之间淡入淡出并进行转换的过程，apply方法用于捕获对象内容的初始显示，为转换做必要的准备。timeout指定了转换的延迟时间，再加上滤镜中设置的转换时间，则图片在转换之间将停留，以方便清楚地浏览图片。最后，在主体元素body中插入onload事件，加载转换过程。

在IE9.0中浏览效果如图11-3所示，可以看到一张图片慢慢消失，一张图片慢慢出现，两张图片不断循环往复，从而实现渐变效果。

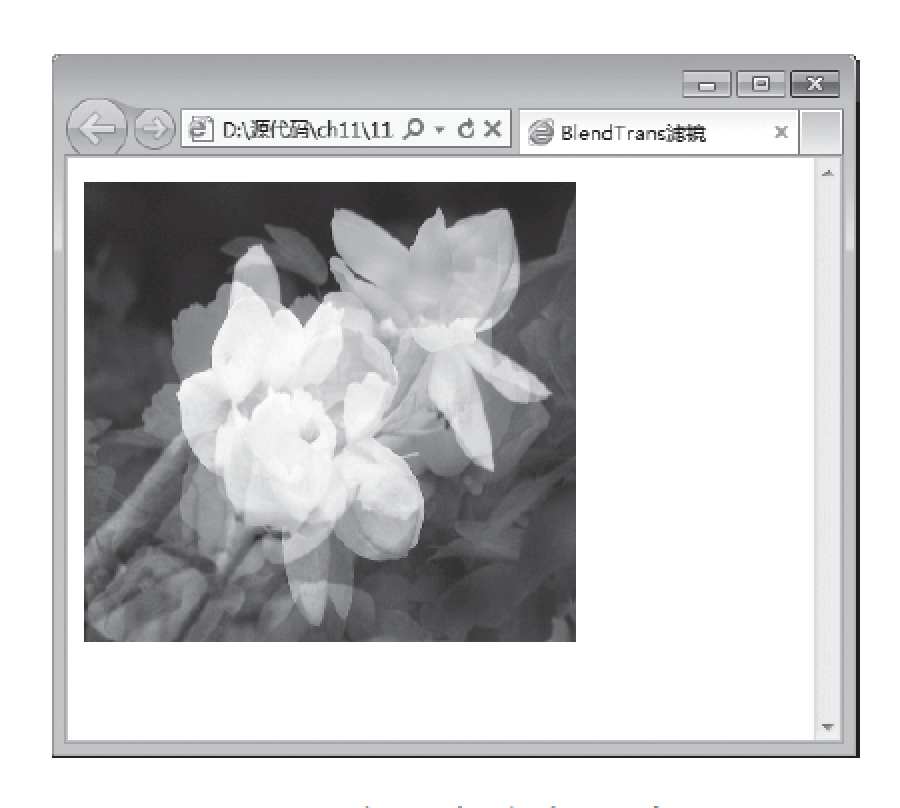


图11-3 实现渐隐渐现效果



11.4 Blur滤镜

Blur滤镜实现页面模糊效果，即在一个方向上的运动模糊。如果应用得当，就可以产生高速移动的动感效果。

Blur滤镜的语法格式如下：

```
{filter : Blur ( enabled=bEnabled , add=iadd , direction=idirection ,
           strength=fstrength )}
```

Blur属性值及其含义如表11-3所示。

表11-3 Blur属性值

参数	说明
enabled	设置滤镜是否激活
add	指定图片是否改变成模糊效果。这是个布尔参数，有效值为True或False。True是默认值，表示应用模糊效果，False则表示不应用
direction	设定模糊方向。模糊的效果是按顺时针方向起作用的，取值范围为0~360°，45°为一个间隔。有8个方向值：0表示向上，45表示右上，90表示向右，135表示右下，180表示向下，225表示左下，270表示向左，315表示左上
strength	指定模糊半径大小，单位是像素，默认值为5，取值范围为自然数，该取值决定了模糊效果的延伸范围

【例11.4】实例文件：ch11\11.4.html

```
<html>
<head>
<title>模糊Blur</title>
<style>
img{
      height:180px;
}
 div.div2 { width:400px;filter:blur(add=true,direction=90,strength=50) }
</style>

</head>
<body>
        原始图<img src="baihua.jpg">
        add=true<img src="baihua.jpg" style="filter:Blur(add=true,direction
=225,strength=20)">
        add=false<img src="baihua.jpg" style="filter:Blur(add=false,
direction=225,strength=20)">
 <div class="div2">
        <p style="font-size: 30pt; font-weight: bold; color:DarkBlue">
          Blur滤镜</p>
```

```
    </div>
</body>
</html>
```

在IE9.0中浏览效果如图11-4所示，可以看到两张模糊图片，在一定方向上发生模糊，下方的文字也发生了模糊，具有文字吹风的效果。

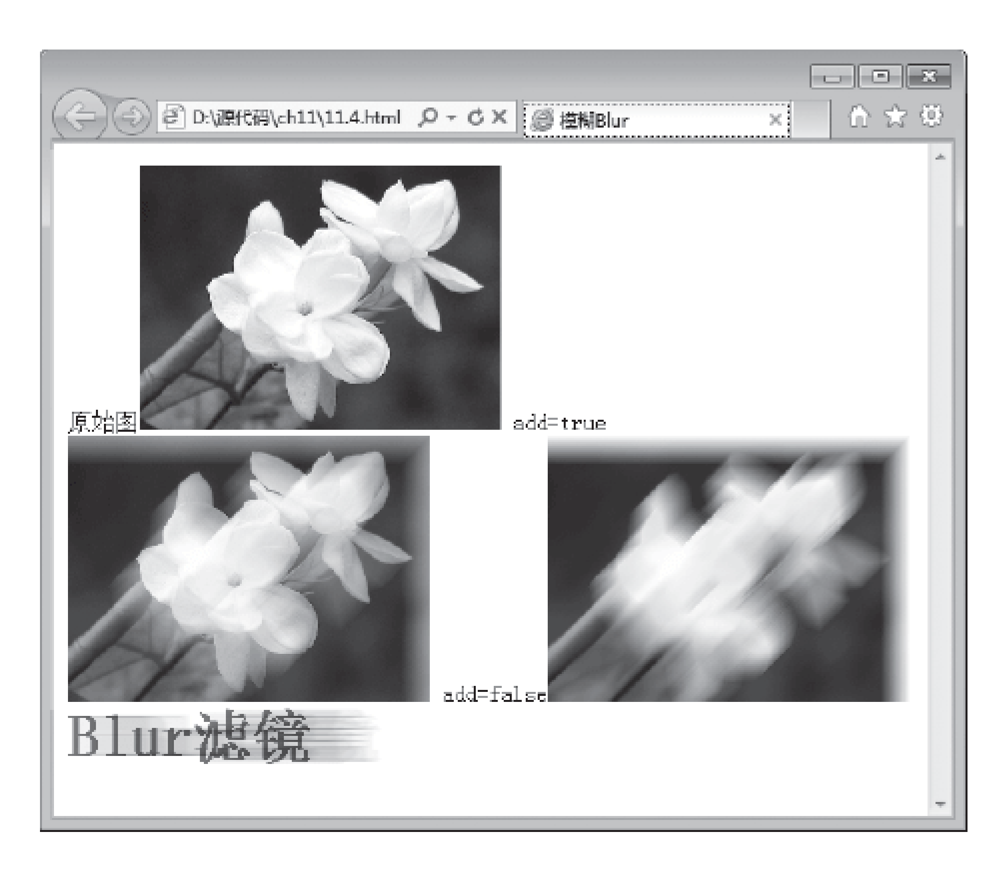


图11-4 模糊效果



11.5 Chroma滤镜

Chroma滤镜可以设置HTML对象中指定的颜色为透明色。其语法格式如下：

```
{filter : Chroma(enabled=bEnabled , color=sColor)}
```

其中，color参数设置要变为透明色的颜色。

【例11.5】实例文件：ch11\11.5.html

```
<html>
<head>
    <title>Chroma滤镜</title>
    <style>
      <!--
        div{position:absolute;top:70;letf:40; filter:Chroma(color=blue)}
        p{font-size:30pt; font-weight:bold; color:blue}
      -->
    </style>
</head>
<body>
    <p>Chroma滤镜效果</p>
```

```
        <div>
            <p>Chroma滤镜效果</p>
        </div>
    </body>
    </html>
```

在IE9.0中浏览效果如图11-5所示，可以看到第二个段落，某些笔画丢失。但拖动鼠标选择过滤颜色后的文字，便可以查看过滤掉颜色的文字。



图11-5 Chroma滤镜效果

Chroma滤镜一般应用于文字特效，而且对于有些格式的图片也是不适用的。例如，JPEG格式的图片是一种已经减色和压缩处理的图片，要设置其中某种颜色透明将十分困难。



11.6　DropShadow滤镜

阴影效果在实际的文字和图片中非常实用，IE9.0通过DropShadow滤镜建立阴影效果，使元素内容在页面上产生投影，从而实现立体的效果。其工作原理就是创建一个偏移量，并定义一个阴影颜色，使之产生效果。

DropShadow滤镜语法格式如下：

```
    {filter:DropShadow ( enabled=bEnabled , color=sColor , offx=iOffsetx, offy=iOffsety,
                    positive=bPositive ) }
```

DropShadow的属性值及其说明如表11-4所示。

表11-4 DropShadow属性值

属性值	说明
enabled	设置滤镜是否激活
color	指定滤镜产生的阴影颜色
offx	指定阴影水平方向偏移量，默认值为5px
offy	指定阴影垂直方向偏移量，默认值为5px
positive	指定阴影透明程度，为布尔值。True（1）表示为任何非透明像素建立可见的阴影；False（0）表示为透明的像素部分建立透明效果

【例11.6】实例文件：ch11\11.6.html

```
    <html>
    <head>
        <title>DropShadow滤镜</title>
    </head>
    <body>
        <table width="90%" height="90%">
            <tr>
                <td style="filter: DropShadow(color=gray,offx=10,offy=10,
positive=1)">
                    <img src="9.jpg" >
                </td>
            </tr>
            <tr>
                <td style="filter: DropShadow(color=gray,offx=5,offy=
5,positive=1);
                         font-size:20pt; color:DarkBlue">
                   这是一个阴影效果
                </td>
            </tr>
        </table>
    </body>
    </html>
```

在IE9.0中浏览效果如图11-6所示，可以看到图片产生了阴影，但不明显，下方文字产生阴影效果明显。



图11-6 阴影效果



FlipH滤镜

在CSS3中，可以通过FlipH滤镜实现HTML对象翻转效果，其中FlipH滤镜用于水平翻转对象，即将元素对象按水平方向进行180° 翻转。FlipH滤镜可以在CSS中直接使用，使用格式如下：

```
{Fliter: FlipH(enabled=bEnabled)}
```

该滤镜中只有一个enabled参数，表示是否激活该滤镜。

【例11.7】实例文件：ch11\11.7.html

```
<html >
<head>
    <title>FlipH滤镜</title>
<style>
img{
height:120px;
width:200px;
}
</style>
</head>
<body>
        原图片<img src="9.jpg">
        图片水平翻转<img src="9.jpg" style="Filter:FlipH()">

</body>
</html>
```

在IE9.0中浏览效果如图11-7所示，可以看到图片以中心为支点，进行了左右方向上的翻转。



图11-7　图片水平翻转



11.8 Glow滤镜

文字或物体发光的特性往往能吸引浏览者注意，Glow滤镜可以使对象的边缘产生一种柔和的边框或光晕，并可产生如火焰一样的效果。

语法格式如下：

```
{filter : Glow ( enabled=bEnabled , color=sColor , strength=iDistance ) }
```

其中，color设置边缘光晕颜色，strength设置晕圈范围，值范围是1~255，值越大效果越强。

【例11.8】实例文件：ch11\11.8.html

```
<html>
<head>
    <title>filter glow</title>
    <style>
    <!--
      .weny{
            width:100%;
            filter:Glow(color=#9966CC,strength=10)}
    -->
    </style>
</head>
<body>
    <div class="weny">
        <p style="font-family: 1幼圆; font-size: 40pt; font-weight:
bolder; color: #003366">
            这段文字带有光晕，喜欢么</p>
    </div>
</body>
</html>
```

在IE9.0中浏览效果如图11-8所示，可以看到文字带有光晕出现，非常漂亮。

当Glow滤镜作用于文字时，每个文字边缘都会出现光晕，效果非常强烈。而对于图片，Glow滤镜只在其边缘加上光晕。



图11-8 光晕效果



11.9　Gray滤镜

黑白色是一种经典颜色，使用Gray滤镜能够轻松地将彩色图片变为黑白图片。其代码格式如下：

```
{filter:Gray(enabled=bEnabled)}
```

enabled表示是否激活，可以在页面代码中直接使用。

【例11.9】实例文件：ch11\11.9.html

```
<html>
<head>
<title>Gray滤镜</title>
</head>
<body>
          <img src="9.jpg"   style="width: 50%;height:50%"  />原图
           <img src="9.jpg"  style="width: 50%;height:50%; filter: Gray()"
/>  灰度图
</body>
</html>
```

在IE9.0中浏览效果如图11-9所示，可以看到下面一张图片以黑白色显示。



图11-9　黑白色显示图片



11.10 Invert滤镜

Invert滤镜可以把对象的可视化属性全部翻转，包括色彩、饱和度和亮度值，使图片产生一种“底片”或负片的效果。其语法格式如下：

```
{filter:Invert(enabled=bEnabled)}
```

enabled参数用来设置是否激活滤镜。

【例11.10】实例文件：ch11\11.10.html

```
<html>
<head>
<title>Invert滤镜</title>
</head>
<body>
<img src="9.jpg" />原图
<img src="9.jpg"  style="width:30%; filter: Invert()" />反相图
</body>
</html>
```

在IE9.0中浏览效果如图11-10所示，可以看到下面一张图片以相片底片的颜色出现。

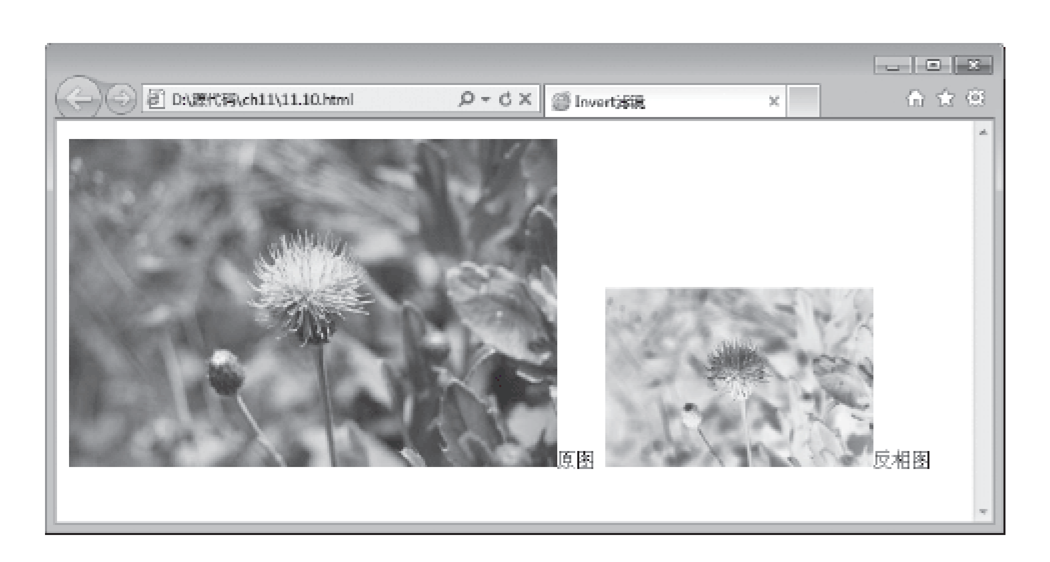


图11-10 反色效果



11.11 FlipV滤镜

FlipH滤镜用来实现对象的水平翻转，而FlipV滤镜用来实现对象的垂直翻转，其中包括文字和图像。其语法格式如下：

```
{Fliter: FlipV(enabled=bEnabled)}
```

enabled参数表示是否激活滤镜。

【例11.11】实例文件：ch11\11.11.html

```
<html>
<head>
<title>FlipV滤镜</title>
</head>
<body>
        <img src="9.jpg">原图片
        <img src="9.jpg" style="Filter:FlipV()">图片垂直翻转
</body>
</html>
```

在IE9.0中浏览效果如图11-11所示，可以看到右方图片上下发生了翻转。

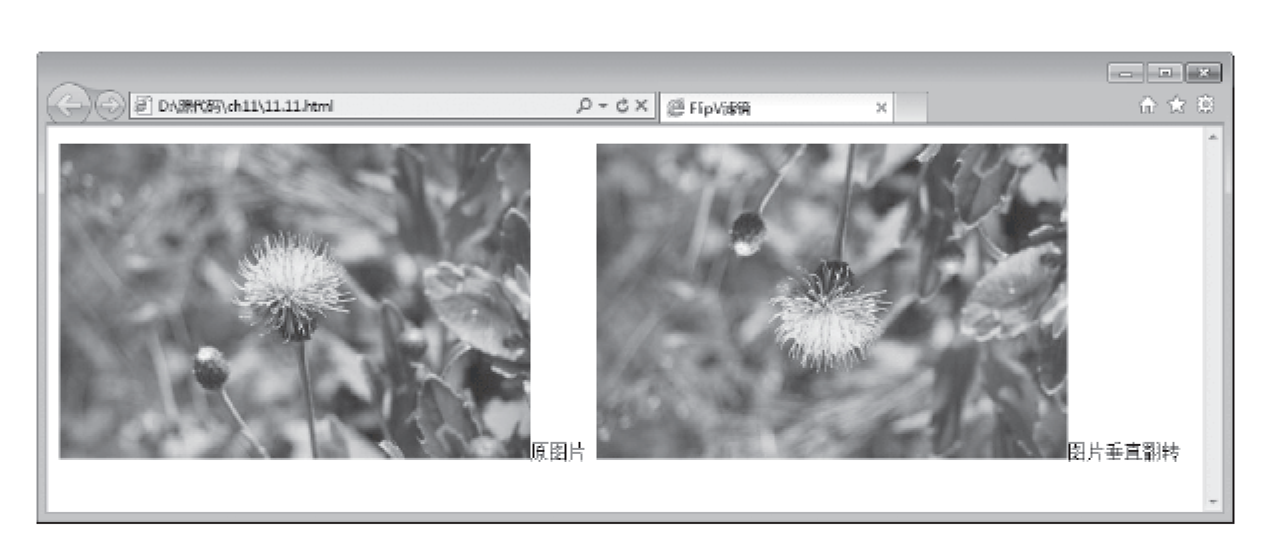
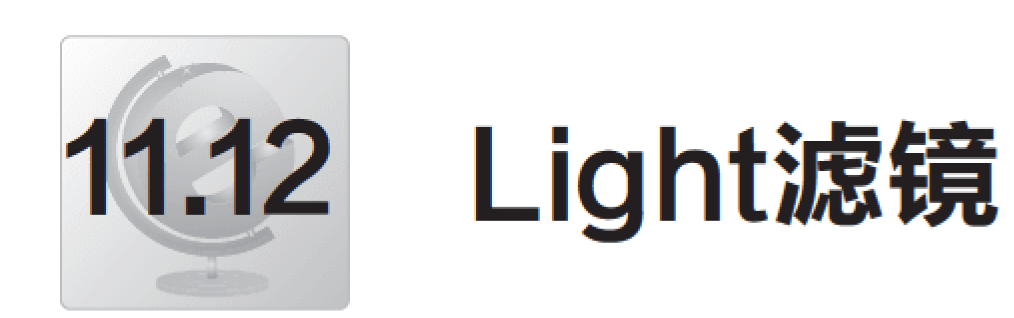


图11-11　上下翻转



11.12　Light滤镜

Light滤镜是一个高级滤镜，需要结合JavaScript使用。该滤镜用来产生类似于光照灯效果，并调节亮度以及颜色。其语法格式如下：

```
{filter:Light(enabled=bEnabled)}
```

对于已定义的Light滤镜属性，可以调用它的方法(Method)来设置或改变属性。这些方法如表11-5所示。

表11-5　Light滤镜使用方法

方法	说明
AddAmbIE9.0nt	加入包围的光源
AddCone	加入锥形光源
AddPoint	加入点光源
Changcolor	改变光的颜色
Changstrength	改变光源的强度
Clear	清除所有的光源
MoveLight	移动光源

【例11.12】实例文件：ch11\11.12.html

```
<html>
<head>
<title>light滤镜效果</title>
</head>
<body>
    <table>
        <tr>
            <td style="color:blue; font-weight:bolder">
                 随鼠标变化的动态光源效果
            </td>
        </tr>
        <tr>
            <td id="light" style="filter: light(); width: 200px">
                <img src="feng.jpg">
            </td>
        </tr>
    </table>
    <script language="Javascript">
    <!--
       var g_numlights=0;
       // 调用设置光源函数
       window.onload=setlights;
       // 获得鼠标句柄
       light.onmousemove=mousehandler;
       //建立光源的集合
       function setlights(){
          light.filters[0].clear();
          light.filters[0].addcone(0,0,5,100,100,255,255,0,60,30);
       }
       // 捕捉鼠标的位置来移动光线焦点
       function mousehandler(){
          x=(window.event.x-80);
          y=(window.event.y-80);
          light.filters[0].movelight(0,x,y,5,1);
       }
    -->
    </script>
</body>
</html>
```

在IE9.0中浏览效果如图11-12所示，可以看到一幅图片实现光照的效果，而且随着鼠标的移动，灯光照射的方向也不相同，类似于镭射灯的效果。

要实现光照效果，JavaScript脚本语言起主要作用。首先要创建光源并指定光源位置。setlights函数中filters[0]表示设置的光源滤镜，调用clear方法在每次页面加载时先清除所有的光源，然后再使用addcone方法创建锥形光源。如果需要在图片上添加多束光源，则可以重复使用addcone方法，但注意要使用不同的参数，否则光源处于同一位置，就无法产生效果了。函数mousehandler用来实现光束随着鼠标移动的效果。



图11-12 Light滤镜效果



Mask滤镜

可以通过遮罩mask，为网页中的元素对象做出一个矩形遮罩。遮罩就是使用一个颜色图层将包含有文字或图像等对象的区域遮盖，但是文字或图像部分却以背景色显示出来。

Mask滤镜语法格式如下：

```
{filter:Mask(enabled=bEnabled , color=sColor)}
```

参数color用来设置Mask滤镜作用的颜色。

【例11.13】实例文件：ch11\11.13.html

```
<html>
<head>
<title>Mask遮罩滤镜</title>
<style>
p {
    width:400;
filter:mask(color:#FF9900);
    font-size:40pt;
    font-weight:bold;
    color:#00CC99;
}
</style>
</head>
<body>
<p>这里有个遮罩</p>
</body>
</html>
```

在IE9.0中浏览效果如图11-13所示，可以看到文字上面有一个遮罩，文字颜色是背景颜色。

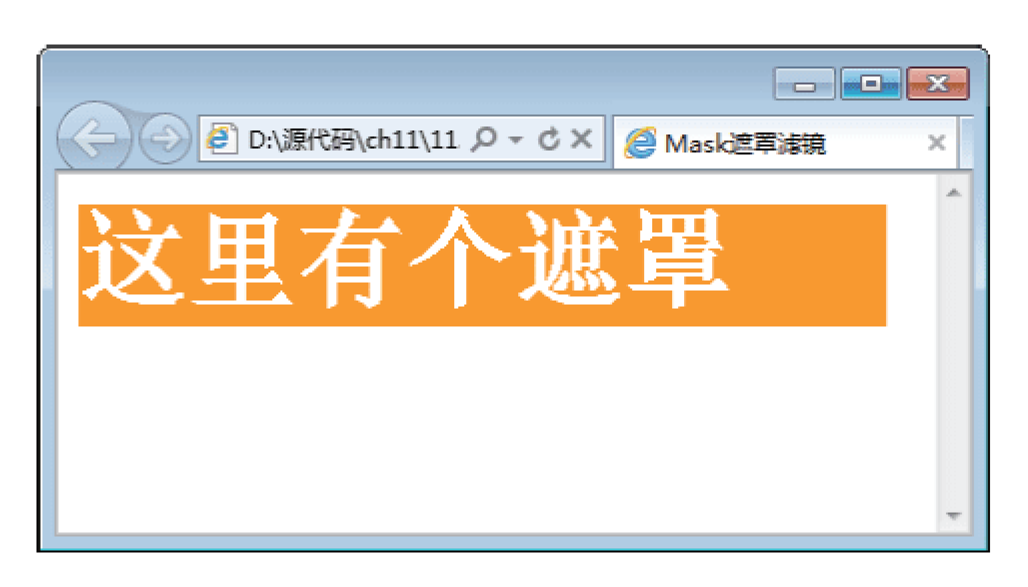


图11-13 文字遮罩



11.14 RevealTrans滤镜

RevealTrans滤镜能够实现图像之间的切换效果。切换时，能产生32种动态效果，例如溶解、水平（垂直）展幕、百叶窗等，而且还可以随机选取其中一种效果进行切换。

RevealTrans滤镜语法格式如下：

```
    filter : RevealTrans ( enabled=bEnabled , duration=fDuration ,
transition=iTransitionType )
```

其中，enabled表示是否激活滤镜；duration用于设置切换停留时间；transition用于指定转换方式，即指定要使用的动态效果，参数取值是0~23。

transition参数值如表11-6所示。

表11-6 RevealTrans滤镜动态效果

动态效果	参数值	动态效果	参数值
矩形从大至小	0	随机溶解	12
矩形从小至大	1	从上下向中间展开	13
圆形从大至小	2	从中间向上下展开	14
圆形从小至大	3	从两边向中间展开	15
向上推开	4	从中间向两边展开	16
向下推开	5	从右上向左下展开	17
向右推开	6	从右下向左上展开	18
向左推开	7	从左上向右下展开	19
垂直形百叶窗	8	从左下向右上展开	20
水平形百叶窗	9	随机水平细纹	21
水平棋盘	10	随机垂直细纹	22
垂直棋盘	11	随机选取一种效果	23

但是，如果只设置了transition参数来实现切换过程的话，那么是不会有任何效果的，因

为动态效果的实现还必须依靠脚本语言JavaScript调用相应的方法。

【例11.14】实例文件：ch11\11.14.html

```
<html >
<head>
    <title>BlendTrans滤镜</title>
<style type="text/css">
  .revealtrans { filter:revealTrans(Transition=10,Duration=3)}
</style>
</head>
<body onload="playImg()">
  <img id="imgpic" class="revealtrans" src="9.jpg"/>
  <script language="JavaScript">
  <!--
    // 声明数组，数组元素的个数就是图片的个数，然后给数组元素赋值，值为图片路径
    ImgNum=new ImgArray(2);
    ImgNum[0]="baimd.jpg";
    ImgNum[1]="baihua.jpg";
    // 获取数组记录数
    function ImgArray(len)
    {
      this.length=len;
    }
    var i=1;
    //转换过程
    function playImg(){
      if (i==1){
        i=0 ;
      }
      else{
        i++;
      }
      imgpic.filters[0].apply();
      imgpic.src=ImgNum[i];
      imgpic.filters[0].play();
      // 设置演示时间，这里是以毫秒为单位的，4000则表示延迟秒
      // 滤镜中设置的转换时间值，这样当转换结束后还停留一段时间
      timeout=setTimeout('playImg()',4000);
    }
    -->
    </script>
</body>
</html>
```

对于图片切换，实现代码与BlendTrans滤镜的转换代码相似，这里就不再介绍了。

在IE9.0中浏览效果如图11-14所示，可以看到图片以百叶窗的形式打开另一个图片，如此循环往复。

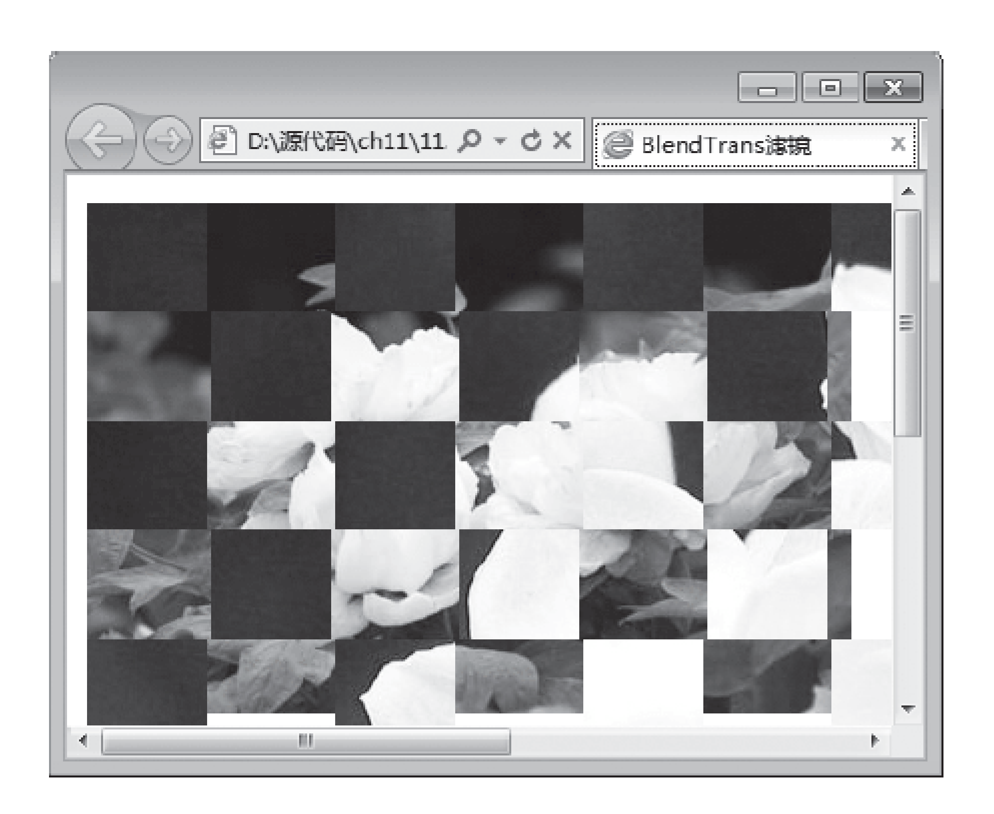


图11-14 图片切换效果



11.15 Shadow滤镜

可以通过Shadow滤镜来给对象添加阴影效果，其实际效果看起来好像是对象离开了页面，并在页面上显示出该对象阴影。阴影部分的工作原理是建立一个偏移量，并为其加上颜色。

语法格式如下：

```
{filter:Shadow(enabled=bEnabled , color=sColor , direction=iOffset,
strength=iDistance)}
```

Shadow的各参数如表11-7所示。

表11-7 Shadow滤镜参数

参数	说明
enabled	设置滤镜是否激活
color	设置投影的颜色
direction	设置投影的方向，分别有8种取值代表8种方向：取值为0表示向上，45为右上，90为右，135为右下，180为下方，225为左下方，270为左方，315为左上方
strength	设置投影向外扩散的距离

【例11.15】实例文件：ch11\11.15.html

```
<html>
<head>
<title>阴影效果</title>
<style>
h1 {
    color:#FF6600;
    width:400;
```

```
    filter:shadow(color=blue, offx=15, offy=22, positive=false);
}
</style>
</head>
<body>
<h1>我好看么</h1>
</body>
</html>
```

在IE9.0中浏览效果如图11-15所示，可以看到文字带有阴影效果。

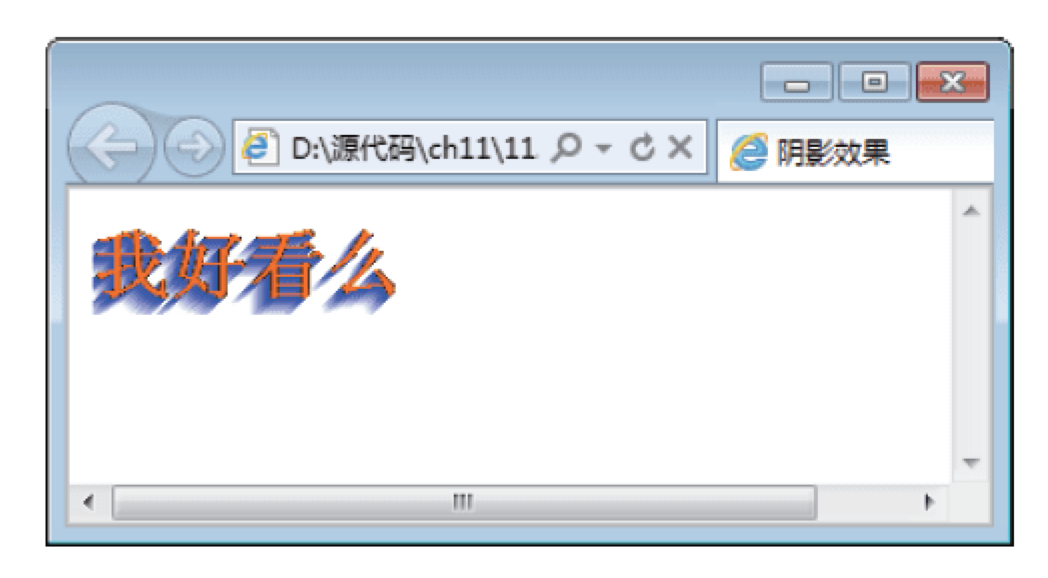


图11-15 Shadow滤镜效果



11.16 Wave滤镜

Wave滤镜可以为对象添加竖直方向上的波浪效果，也可以用来把对象按照竖直的波纹样式打乱。

语法格式如下：

```
{filter:Wave (enabled=bEnabled , add=bAddImage , freq=iWaveCount ,
lightStrength=iPercentage , phase=iPercentage , strength=iDistance) }
```

Wave的各参数说明如表11-8所示。

表11-8 Wave滤镜参数

参数	说明
enabled	设置滤镜是否激活
add	布尔值，表示是否在原始对象上显示效果。True表示显示，False表示不显示
freq	设置生成波纹的频率，也就是设定在对象上产生的完整的波纹的条数
lightStrength	波纹效果的光照强度，取值0~100
phase	设置正弦波开始的偏移量，取百分比值为0~100，默认值为0。25就是360° ×25%为90° ，50则为180°
strength	波纹曲折的强度

【例11.16】实例文件：ch11\11.16.html

```
<html>
<head>
<title>波浪效果</title>
<style>
h1 {
    color:violet;
    text-align:left;
    width:400;
    filter:wave(add=true, freq=5, lightStrength=45, phase=20, strength=3);
}
</style>
</head>
<body>
<h1>一起去看大海</h1>
</body>
</html>
```

在IE9.0中浏览效果如图11-16所示，可以看到文字带有波浪效果。

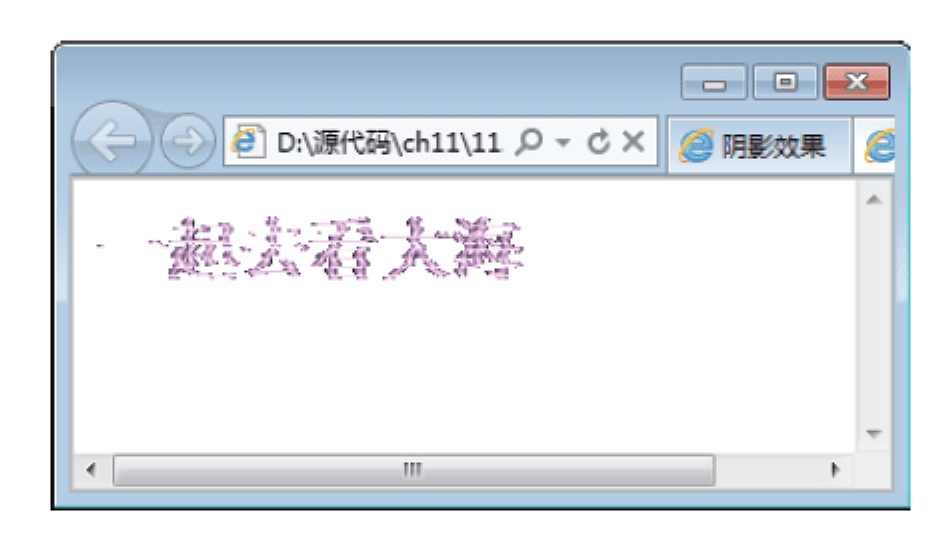


图11-16 波浪效果

 X-ray滤镜

X-ray中文含义为X射线，X-ray滤镜可以使对象反映出它的轮廓，并把这些轮廓的颜色加亮，使整体看起来会有一种X光片的效果。

语法格式如下：

```
{filter:Xray(enabled=bEnabled)}
```

enabled参数用于确定是否激活该滤镜。

【例11.17】实例文件：ch11\11.17.html

```
<html>
<head>
```

```
<title>X射线</title>
<style>
.noe {
filter:xray;
}
</style>
</head>
<body>
<img src="9.jpg" class="noe" />
<img src="9.jpg" />
</body>
</html>
```

在IE9.0中浏览效果如图11-17所示，可以看到图片有X光滤镜效果。

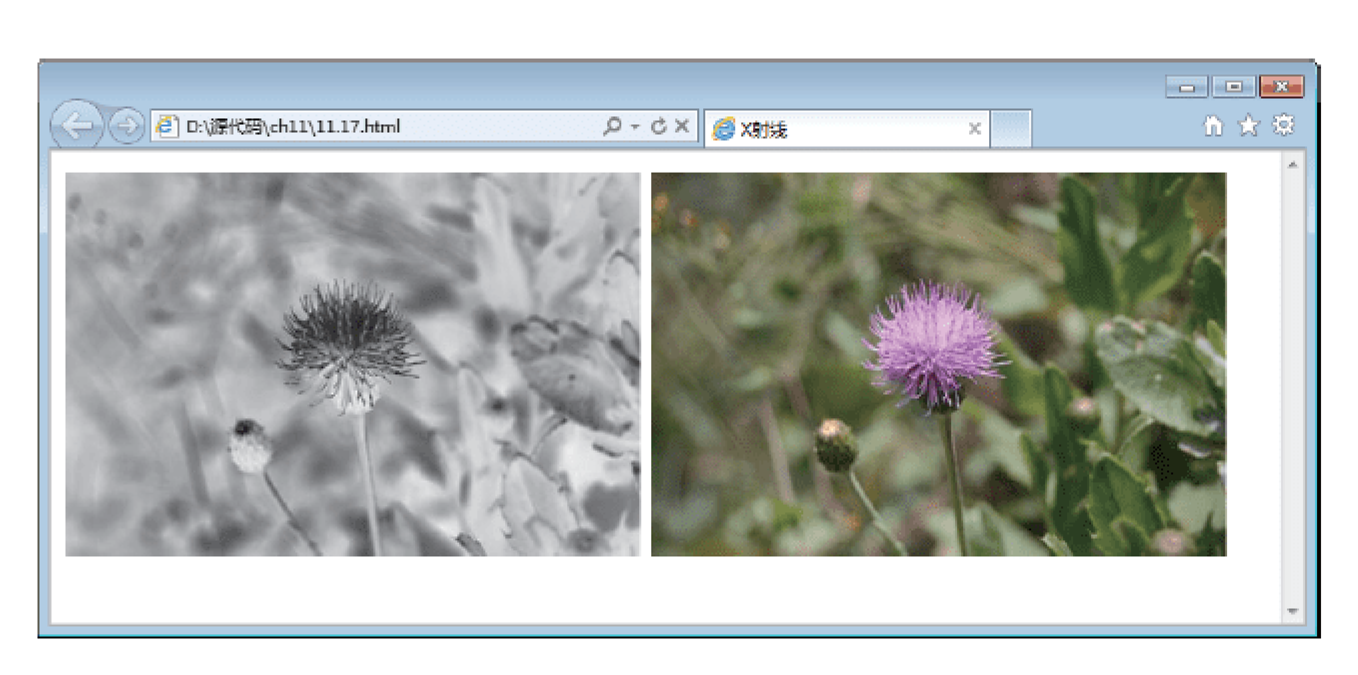


图11-17　X滤镜效果



11.18 专家解惑

1．如何对一个HTML对象使用滤镜效果？

在使用滤镜时，若使用多个滤镜，则每个滤镜之间用空格分隔开；一个滤镜中的若干个参数用逗号分隔；filter属性和其他样式属性并用时以分号分隔。

2．滤镜效果是IE9.0浏览器特有的CSS特效，那么在Firefox中能不能实现呢？

滤镜效果虽然是IE9.0浏览器特有的效果，但使用Firefox浏览器一些属性也可以实现相同效果。例如IE9.0的阴影效果，可以在Firefox网页设计中，先在文字下面再叠一层浅色的同样的字，然后做2px的错位，制造阴影的假象。



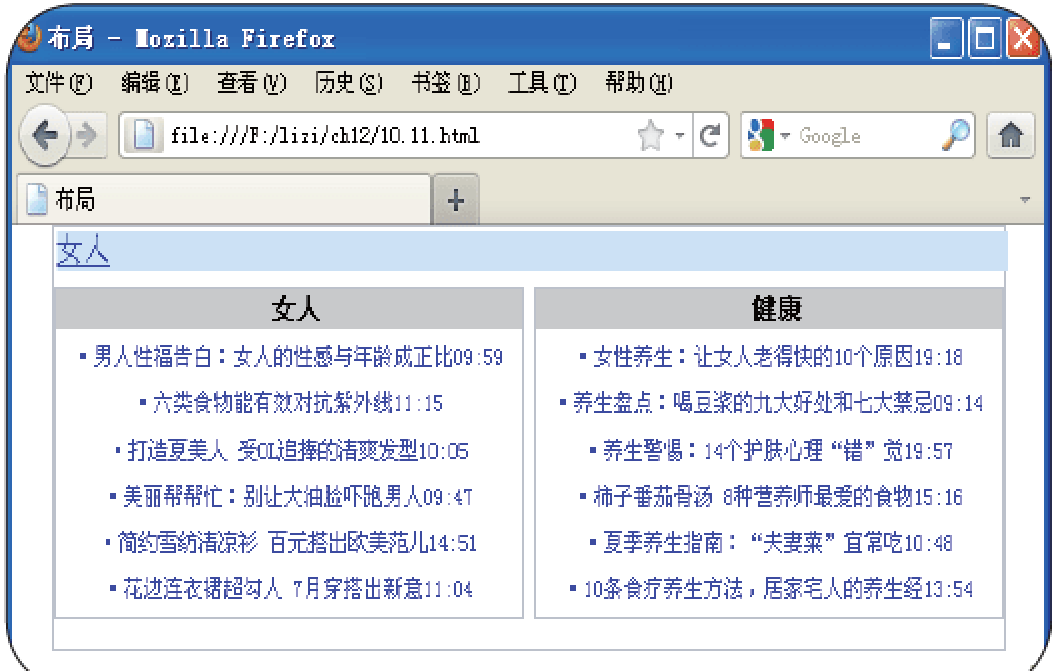
第 12 章

定位与布局

本章引言

网页设计中，能否很好地定位网页中的每个元素，是网页整体布局的关键。一个布局混乱、元素定位不准的页面，是每个浏览者都不喜欢的。而把每个元素都精确定位到合理位置，才是构建美观大方页面的前提。





 定位方式

在CSS3中，定位可以将一个元素精确地放在页面上用户所指定的位置，而布局是将整个页面的元素内容整洁且完美地摆放。定位的实现是布局成功的前提。如果清晰地掌握了定位原理，那么就能够创建多种高级而精确的布局，并会让网页更加完美地实现。

12.1.1 定位属性

在网页设计中，定位（Positioning）思想简单，就是用户精确地定义HTML元素框所在页面中的位置，可以是页面绝对位置，也可以处于其上级元素、另一个元素或浏览器窗口的绝对位置。

元素框是指包含每个元素的矩形框。而元素内容与元素框共同形成了元素块。定位是指设定元素块位置和大小。实现CSS3定位，需要依赖定位属性才能够完成。

表12-1列出了CSS3中全部有关的定位属性。

表12-1 定位属性

属性	含义
position	定义位置
left	指定元素横向距左部距离
right	指定元素横向距右部距离
top	指定元素纵向距顶部距离
bottom	指定元素纵向距底部距离
z-index	设置元素的层叠顺序
width	设置元素框宽度
height	设置元素框高度
overflow	内容溢出控制
clip	剪切

表中前6个属性是实际的定位属性，后面的4个有关属性，是用来设置元素框，或对元素框中的内容进行控制的属性。其中，position属性是最主要的定位属性。它既可以定义元素框的绝对位置，又可以定义相对位置，而left、right、top和bottom只在position属性中使用才会发挥作用。

12.1.2 position定位

网页中各种元素需要经过合理地定位来搭建整个页面的结构。在CSS3中，可以通过position这个属性，对页面中元素进行定位。

语法格式如下：

```
position : static | absolute | fixed | relative
```

position属性值及其含义如表12-2所示。

表12-2 position属性值

属性值	说明
static	元素定位的默认值，无特殊定位，对象遵循HTML定位规则，不能通过z-index进行层次分级
relative	相对定位，对象不可重叠，可以通过left、right、bottom和top等属性在正常文档中偏移位置，可以通过z-index进行层次分级
absolute	生成绝对定位的元素，相对于static定位以外的第一个父元素进行定位。元素的位置通过left、top、right和bottom属性进行规定
fixed	fixed生成绝对定位的元素，相对于浏览器窗口进行定位。元素的位置通过left、top、right和bottom属性进行规定

1．绝对定位absolute

绝对定位是参照浏览器的左上角，配合top、left、bottom和right进行定位的，如果没有设置上述的4个值，则默认依据父级的坐标原点为原始点。绝对定位可以通过上、下、左、右来设置元素，使其处在任何一个位置。

在父层position属性为默认值时，上、下、左、右的坐标原点以body的坐标原点为起始位置。绝对定位的语法格式如下：

```
position:absolute
```

只要将上面代码加入到样式中，使用样式的元素就可以以绝对定位的方式显示了。

【例12.1】实例文件：ch12\12.1.html

```
<html>
<head>
<title>定位属性</title>
</head>
<body>
  <div style="background-color: Black; width:200px; height:200px">
    <h2 style=" position:absolute; left:80px; top:80px; width:110px;
height:50px;
        background-color:Red;">这是绝对定位</h2>
  </div>
</body>
</html>
```

在IE9.0中浏览效果如图12-1所示，可以看到红色元素框依据浏览器左上角为原点，坐标位置为（80px，80px），宽度为110px，高度为50px。



图12-1 绝对定位

提示　使用绝对定位会产生一个问题：目前，大多数的网页都是居中显示的，且元素与元素之间布局是紧密的，而绝对定位的开始位置是浏览器的左上角的0点，当设定各元素块边偏移属性时，由于客户端屏幕分辨率的不同，各元素块的显示可能会有偏差。这是由于页面的显示是随着分辨率的大小而自动适应，而各元素块在参照绝对定位的位置显示，那么在浏览器的视野范围内原始页面可以或超出或缩小地显示。

优秀的页面设计能够适用各种屏幕分辨率，并且保证正常显示。要解决这个问题，在定位时最好使用相对定位。

2. 相对定位relative

如果对一个元素进行相对定位，首先它将出现在它所在的位置上。然后通过设置垂直或水平位置，让这个元素“相对于”它的原始起点进行移动。另外，相对定位时，无论是否进行移动，元素仍然占据原来的空间。因此，移动元素会导致它覆盖其他框。

绝对定位与相对定位的区别在于：绝对定位的坐标原点为上级元素的原点，与上级元素有关；相对定位的坐标原点为本身偏移前的原点，与上级元素无关。

相对定位的语法格式如下：

```
position:relative
```

【例12.2】实例文件：ch12\12.2.html

```
<html>
<head>
<style type="text/css">
h2.pos_left
{
position:relative;
left:-20px
}
h2.pos_right
{
position:relative;
left:20px
}
</style>
</head>
```

```
<body>
<h2>这是位于正常位置的标题</h2>
<h2 class="pos_left">这个标题相对于其正常位置向左移动</h2>
<h2 class="pos_right">这个标题相对于其正常位置向右移动</h2>
<p>相对定位会按照元素的原始位置对该元素进行移动。</p>
<p>样式 "left:-20px" 从元素的原始左侧位置减去 20 像素。</p>
<p>样式 "left:20px" 向元素的原始左侧位置增加 20 像素。</p>
</body>
</html>
```

在IE9.0中浏览效果如图12-2所示，可以看到页面显示了3个标题，最上面标题正常显示，下面两个标题分别以正常位置为原点，向左或向右分别移动了20像素。

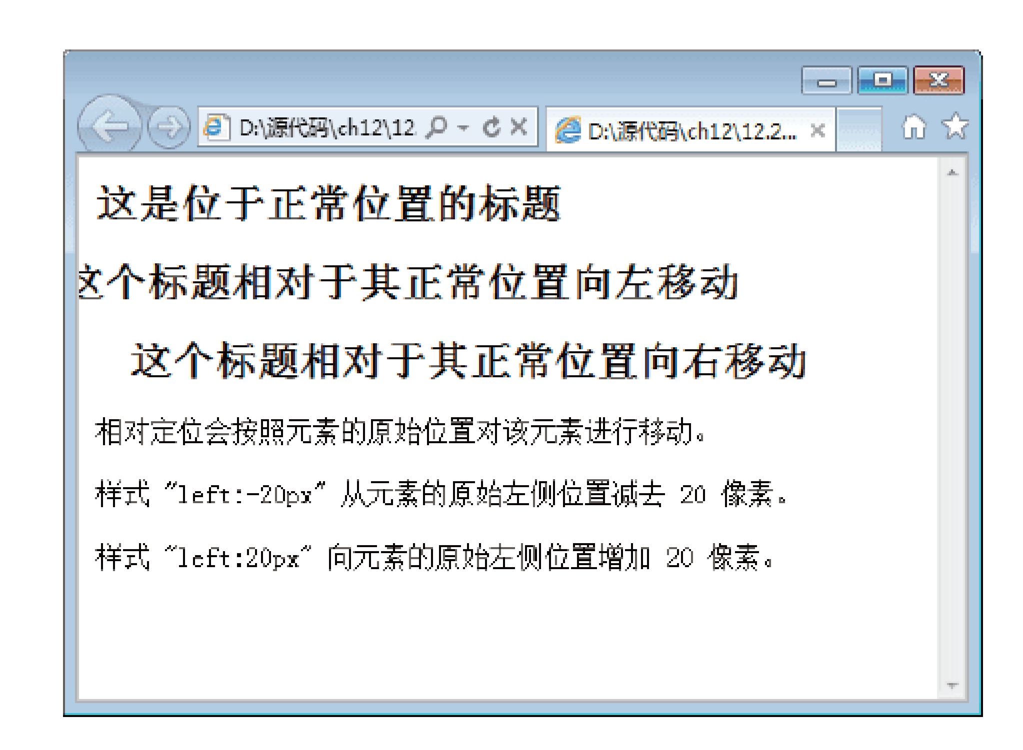


图12-2 相对定位

3．固定定位fixed

固定定位和绝对定位比较相似，它是绝对定位的一种特殊形式，固定定位的容器不会随着滚动条的拖动而变化位置。在视线中，固定定位的容器位置是不会改变的。固定定位可以把一些特殊效果固定在浏览器的视线位置。

固定定位的参照位置不是上级元素块而是浏览器窗口。所以可以使用固定定位来设定类似传统框架样式布局，以及广告框架或导航框架等。使用固定定位的元素可以脱离页面，无论页面如何滚动，始终处在页面的同一位置上。

固定定位语法格式如下：

```
position:fixed
```

【例12.3】实例文件：ch12\12.3.html

```
<html>
<head>
<title>CSS固定定位</title>
<style type="text/css">...
* {
padding:0;
margin:0;
}
#fixedLayer {
width:100px;
```

```
    line-height:50px;
    background: #FC6;
    border:1px solid #F90;
    position:fixed;
    left:10px;
    top:10px;
    }
    </style>
    </head>
    <body>
    <div id="fixedLayer">固定不动</div>
    <p>我动了</p>
    <p>我动了</p>
    <p>我动了</p>
    <p>我动了</p>
    <p>我动了</p>
    <p>我动了</p>
    <p>我动了</p>
    <p>我动了</p>
    <p>我动了</p>
    <p>我动了</p>
    <p>我动了</p>
    <p>我动了</p>
    </body>
    </html>
```

在IE9.0中浏览效果如图12-3所示，可以看到拉到滚动条时，无论页面内容怎么变化，其黄色框“固定不动”始终处在页面左上角顶部。

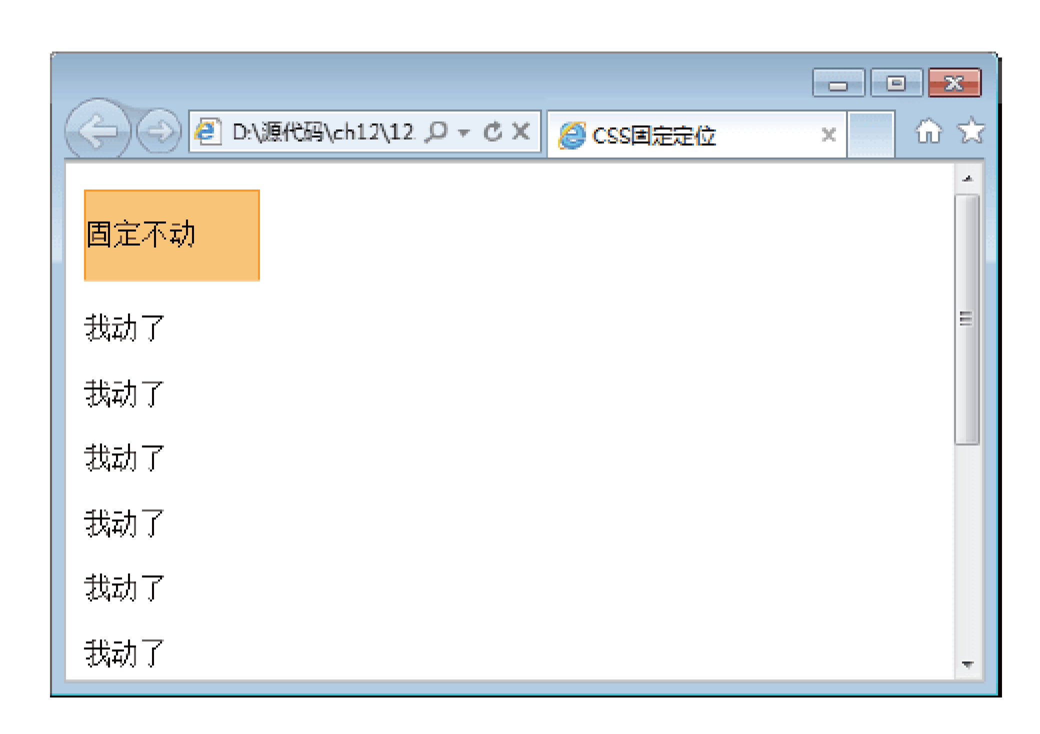


图12-3　固定定位

12.1.3　层叠顺序z-index

对HTML元素进行定位时，可以从其高度、宽度和深度3个方面入手，高度使用height、宽度使用width、深度使用z-index。z-index用来设置元素层叠的次序，其方法是每个元素指定一个数字，数字较大的元素将叠加在数字较小的元素之上。

z-index语法格式如下：

```
z-index : auto | number
```

其参数值auto表示遵循父对象的定位；number是一个无单位的整数值，可以为负值。如果两个决定定位元素的z-index属性具有相同的number值，则依据该元素在HTML文档中声明的顺序进行层叠。如果绝对定位的元素没有指定z-index属性，则此属性的number值为正数的元素会叠加在该元素之上，而number值为负数的对象在该元素之下。如果将参数设置为null，可以消除此属性。该属性只作用于position的属性值为relative或absolute的对象，不能作用在窗口组件上。

【例12.4】实例文件：ch12\12.4.html

```
<html>
<head>
<title>Z-index使用</title>
<style>
#big {
    width:800px;
    height:220px;
    padding:6px;
    background-color:#999999;
    position:relative;
}
#Div1 {
    width:160px;
    height:80px;
    background-color:#FFD700;
    padding:6px;
    position:absolute;
    left:9px;
    top:9px;
    z-index:6;
}
#Div2 {
    width:120px;
    height:80px;
    background-color:thistle;
    padding:6px;
    position:absolute;
              left:280px;
              top:90px;
    z-index:4;
}
#Div3 {
    width:140px;
    height:80px;
    background-color:lightskyblue;
    padding:6px;
    position:absolute;
    left:150px;
    top:25px;
```

```
        z-index:5;
    }
    </style>
    </head>
    <body>
    <div id="big">
      <div id="Div1"><br>
        z-index值是6 ; </div>
      <div id="Div2"><br>
         z-index值是4 ; </div>
      <div id="Div3"><br>
        z-index值是 5 ; </div>
    </div>
    </body>
    </html>
```

在IE9.0中浏览效果如图12-4所示，可以看到网页中显示了3个层，3个层中数值大小不同，并从大到小分别在别的层上显示。

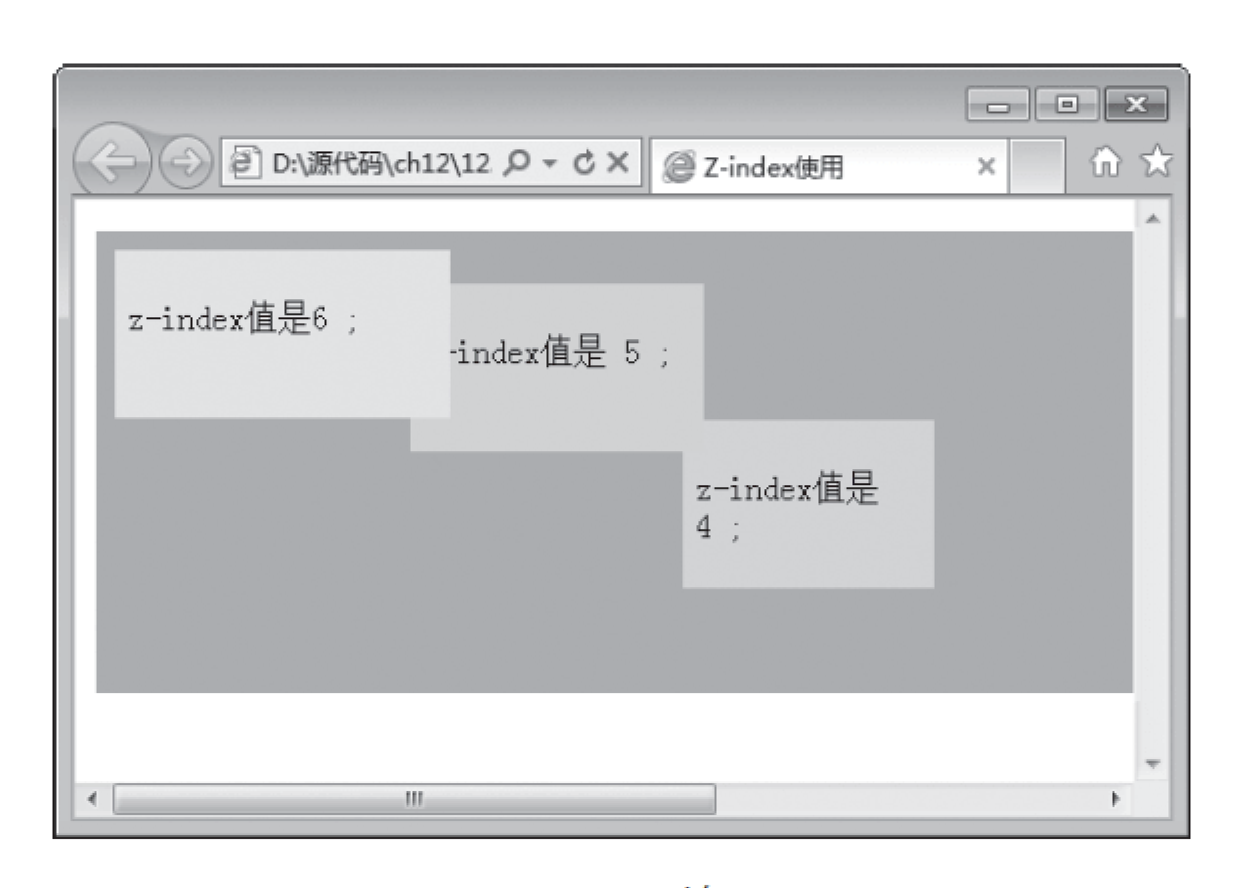


图12-4 z-index效果显示

12.1.4 边偏移属性

边偏移属性包含left、right、top和bottom属性。所谓边偏移属性，就是用来描述元素块与包含元素块最近的边线之间的偏移量的属性。其中left描述元素块最左边与包含其边框最左边的边线的距离，如果left属性值为正，则会偏右移；如果为负，则会使它更向左移，甚至移出边线。其他依次类推。

left、right、top和bottom 4个属性取值非常相似，这里以left为例介绍。left语法格式如下：

```
    left : auto | length
```

上面参数值中auto表示系统自动取值，length表示由浮点数字和单位标识符组成的长度值或百分数。直接设定数值用来设置元素的绝对位置，一旦该位置确定，该元素将始终处于页面中的该位置。使用百分比设置元素位置，是相对于其上级元素的位置而设定的。如果取值为auto，则在定位中允许元素刚好有显示其内容所需的宽度及高度，而不必再指明宽度及高

度的值了。

【例12.5】实例文件：ch12\12.5.html

```
<html>
<head>
<title>定位属性</title>
</head>
<body>
  <div style="background-color: Black; width:200px; height:200px">
    <p style=" position:relative; left:50%; right:0; top:50%; bottom:0;
width:100px; height:100px;
        background-color:Red;">hello
    </p>
  </div>
</body>
</html>
```

在IE9.0中浏览效果如图12-5所示，可以看到红色正方形框在指定位置显示，其底侧和右侧分别与大的矩形框对应。



图12-5 边偏移效果显示

 12.2 float浮动定位

除了使用position进行定位外，还可以使用float定位。float定位只能在水平方向上定位，而不能在垂直方向上定位。float属性表示浮动属性，它用来改变元素块的显示方式。

float语法格式如下：

```
float : none | left |right
```

float属性值如表12-3所示。

表12-3 float属性值

属性值	说明
none	元素不浮动
left	浮动在左边
right	浮动在右边

实际上，使用float可以实现两列布局，也就是让一个元素在左浮动，另一个元素在右浮动，并控制好这两个元素的宽带。

【例12.6】实例文件：ch12\12.6.html

```
<html>
<head>
<title>float定位</title>
<style>
* {
    padding:0px;
    margin:0px;
}
.big {
    width:600px;
                height:100px;
    margin:0 auto 0 auto;
                border:#332533 1px solid;

}
.one {
    width:300px;
               height:20px;
    float:left;
    border:#996600 1px solid;
}
.two {
    width:290px;
               height:20px;
    float:right;
    margin-left:5px;
    display:inline;
    border:#FF3300 1px solid;
}
</style>
</head>
<body>
<div class="big">
  <div class="one">
  <p>非诚勿扰</p>
  </div>
  <div class="two">
  <p>中国达人秀</p>
  </div>
```

```
    </div>
</body>
</html>
```

在IE9.0中浏览效果如图12-6所示，可以看到显示了一个大矩形框，大矩形框中存在两个小的矩形框，并且并列显示。

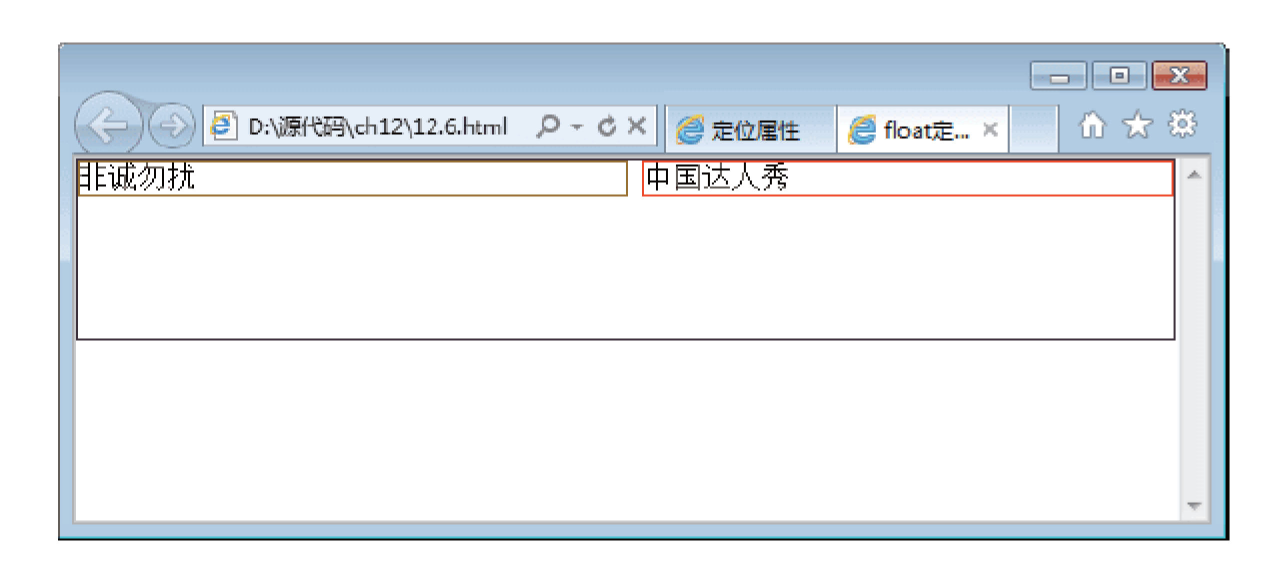


图12-6 float浮动布局

使用float属性不但改变元素的显示位置，同时会对相邻内容造成影响。定义了float属性的元素会覆盖在其他元素上，而被覆盖的区域将处于不可见状态。使用该属性能够实现内容环绕图片的效果。

如果不想让float下面的其他元素浮动环绕在该元素周围，可以使用CSS3属性clear，清除这些浮动元素。

clear语法格式如下：

```
clear : none | left |right | both
```

其中，none表示允许两边都有浮动对象，both表示不允许有浮动对象，left 表示不允许左边有浮动对象，right表示不允许右边有浮动对象。使用float以后，在必要的时候就需要通过clear语句清除float带来的影响，以免出现“其他DIV跟着浮动”的效果。



12.3 overflow溢出定位

如果元素框被指定了大小，而元素的内容不适合该大小，例如元素内容较多而元素框显示不下，此时可以使用溢出属性overflow来控制这种情况。

overflow语法格式如下：

```
overflow : visible | auto | hidden | scroll
```

overflow的各属性值及其说明如表12-4所示。

表12-4　overflow属性值

属性值	说明
visible	若内容溢出，则溢出内容可见
hidden	若内容溢出，则溢出内容隐藏
scroll	保持元素框大小，在框内应用滚动条显示内容
auto	等同于scroll，它表示在需要时应用滚动条

overflow属性适用于以下情况：

- 当元素有负边界时。
- 元素框宽于上级元素内容区，不允许换行。
- 元素框宽于上级元素区域宽度。
- 元素框高于上级元素区域高度。
- 元素定义了绝对定位。

【例12.7】实例文件：ch12\12.7.html

```
<html>
<head>
    <title>overflow属性</title>
    <style >
       div{
           position:absolute;
           color:#445633;
           height:200px;
            width: 30%;
           float:left;
            margin: 0px;
            padding: 0px;
            border-right: 2px dotted #cccccc;
           border-bottom: 2px solid #cccccc;
            padding-right: 10px;
           overflow:auto;
       }
    </style>
</head>
<body >
   <div>
       <p>综艺节目排名</p><p>1 非诚勿扰</p><p>2 康熙来了</p>
       <p>3 快乐大本营</p><p>4 娱乐大风暴</p><p>5 天天向上</p><p>6 爱情连连看</p>
       <p>7 锵锵三人行</p><p>8 我们约会吧</p>
   </div>
</body>
</html>
```

在IE9.0中浏览效果如图12-7所示，可以看到在一个元素框显示了多个元素，拉动显示的滚动条可以查看全部元素。如果overflow设置的值为hidden，则会隐藏多余元素。

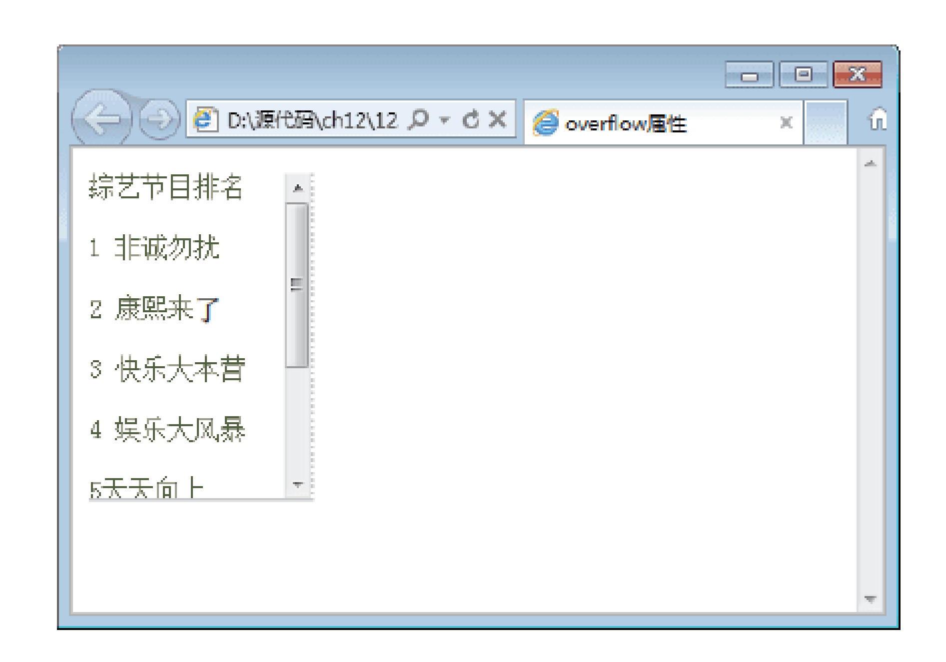


图12-7 溢出定位



12.4 visibility隐藏定位

visibility属性指定是否显示一个元素生成的元素框。这意味着元素仍占据其本来的空间，不过可以完全不可见，即设定元素的可见性。

visibility语法格式如下：

```
visibility : inherit | visible | collapse | hidden
```

其属性值如表12-5所示。

表12-5 visibility属性值

属性值	说明
visible	元素可见
hidden	元素隐藏
collapse	主要用来隐藏表格的行或列。隐藏的行或列能够被其他内容使用。对于表格外的其他对象，其作用等同于hidden

如果元素visibility属性的属性值设定为hidden，表现为元素隐藏，即不可见。但是，元素不可见，并不等同于元素不存在，它仍会占有部分页面位置，影响页面的布局，就如同可见一样。换句话说，元素仍然处于页面中的位置上，只是无法看到它而已。

【例12.8】实例文件：ch12\12.8.html

```
<html>
<head>
  <title>float属性</title>
  <style type="text/css">
    .div{
      padding:5px;
    }
```

```
        .pic{
          float:left;
          padding:20px;
           visibility:visible;
        }
        h1{
           font-weight:bold;
           text-align:center
        }
      </style>
    </head>
    <body>
      <h1>插花</h1>
      <div class="div">
        <div class="pic">
          <img src="08.jpg"  width=150px height=100px />
        </div>
    <p>插花就是把花插在瓶、盘、盆等容器里，而不是栽在这些容器中。所插的花材，或枝、或花、
或叶，均不带根，只是植物体上的一部分，并且不是随便乱插的，而是根据一定的构思来选材，遵循一定
的创作法则，插成一个优美的形体（造型），借此表达一种主题，传递一种感情和情趣，使人看后赏心悦
目，获得精神上的美感和愉快。
    </p>
    <p>
    在我国插花的历史源远流长，发展至今已为人们日常生活所不可缺少。一件成功的插花作品，并不
一定要选用名贵的花材、高价的花器。一般看来并不起眼的一片绿叶、一个花蕾，甚至路边的野花野草和
常见的水果、蔬菜，都能插出一件令人赏心悦目的优秀作品来，使观赏者在心灵上产生共鸣的是创作者唯
一的目的。如果不能产生共鸣，那么这件作品也就失去了观赏价值。具体地说，即插花作品首先要在视觉
上立即引起一种感观和情感上的自然反应，如果未能立刻产生反应，那么摆在眼前的这些花材将无法吸引
观者的目光。在插花作品中引起观赏者情感反应的要素有三点：一是创意（或称立意），指的是表达什么主
题，应选什么花材；二是构思(或称构图)，指的是这些花材怎样巧妙配置造型，在作品中充分展现出各自
的美；三是插器，指的是与创意相配合的插花器皿。三者有机配合，作品便会给人以美的享受。
    </p>
      </div>
    </body>
    </html>
```

在IE9.0中浏览效果如图12-8所示，可以看到图片在左边显示，并被文本信息所环绕。此时visibility属性为visible，表示图片可以看见。



图12-8 隐藏定位显示

12.5 块和行内元素display

在网页设计中，根据需要可以将p段落设置成一个块显示，并带有边框，即类似于DIV层的效果。也可以把多个HTML元素，放在同一行显示。本节将介绍这两种实现方式。

12.5.1 块元素

在CSS3中可以通过display属性控制元素显示，即元素显示方式。

diplay语法格式如下：

```
    display : block | none | inline | compact | marker | inline-table | list-item | run-in | table |table-caption | table-cell | table-column | table-column-group | table-footer-group | table-header-group | table-row | table-row-group
```

display属性值及其含义如表12-6所示。

表12-6 display属性值

属性值	说明
block	以块元素方式显示
inline	以内联元素方式显示
none	元素隐藏
list-item	以列表方式显示
compact	CSS2分配对象为块对象或基于内容之上的内联对象
marker	CSS2指定内容在容器对象之前或之后。要使用此参数，对象必须和:after及:before 伪元素一起使用
inline-table	CSS2将表格显示为无前后换行的内联对象或内联容器
list-item	CSS1将块对象指定为列表项目，并可以添加可选项目标志
run-in	CSS2分配对象为块对象或基于内容之上的内联对象
table	CSS2将对象作为块元素级的表格显示
table-caption	CSS2将对象作为表格标题显示
table-cell	CSS2将对象作为表格单元格显示
table-column	CSS2将对象作为表格列显示
table-column-group	CSS2将对象作为表格列组显示
table-header-group	CSS2将对象作为表格标题组显示
table-footer-group	CSS2将对象作为表格脚注组显示
table-row	CSS2将对象作为表格行显示
table-row-group	CSS2将对象作为表格行组显示

display属性的默认值为block，即元素的默认显示方式是以块元素方式显示。常用的段落p、标题h1、表单form、列表ul和列表选项li都可以定义成块元素。一个块元素，其行高、顶

部和底部都是可控制的。如果不设置宽度的话，块会默认为整个容器的100%；如果设定了值，显示大小由值决定。

【例12.9】实例文件：ch12\12.9.html

```
    <title>块元素</title>
    <style>
        .big{
              width:800px;
              height:105px;
              background-image:url(07.jpg);
              }
        a{
              font-size:12px;
              display:block;
              width:100px;
              height:20px;
              line-height:20px;
              background-color:#F4FAFB;
              text-align:center;
              text-decoration:none;
              border-bottom:1px dotted #6666FF;
                                    color:black;
              }
        a:hover{
              font-size:13px;
              display:block;
              width:100px;
              height:20px;
              line-height:20px;
              text-align:center;
              text-decoration:none;
                                    color:green;
              }
    </style>
    </head>
    <body>
        <div class="big">
              <p>
             <a href="#">管理应用</a><a href="#">财务管理</a><a href="#">在线管理</a>
           <a href="#">客户关系管理</a><a href="#">一体化管理</a>
         </p>
         </div>
    </body>
```

在IE9.0中浏览效果如图12-9所示，可以看到左边显示了一个导航栏，右边显示了一个图片。其导航栏就是以块元素形式显示。

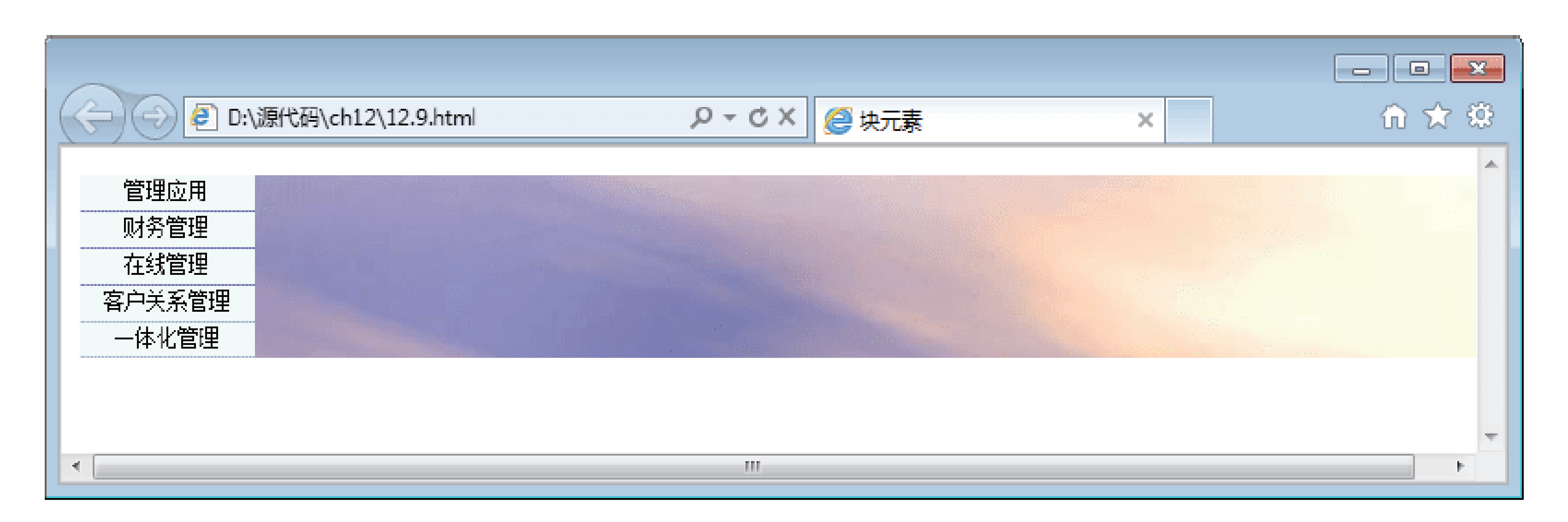


图12-9 块元素显示

12.5.2 行内元素

当display的值被设定为inline时，可以把元素设置为行内元素，并在浏览器中同一行显示。inline元素决定和其他HTML元素在同一行上，其行高、顶部和底部边距可以改变，而宽度是不可以改变的。

【例12.10】实例文件：ch12\12.10.html

```
<html>
<head>
<title>行内元素</title>
<style type="text/css">
.hang {
    display:inline;
}
</style>
</head>
<body>
<div>
<a href="#" class="hang">这是a标签</a>
 <span class="hang">这是span标签</span>
<strong class="hang">这是strong标签</strong>
<img  class="hang" src=6.jpg/>
</div>
</body>
</html>
```

在IE9.0中浏览效果如图12-10所示，可以看到页面显示的4个HTML元素，都在同一行显示，包括超级链接、文本信息和图片。

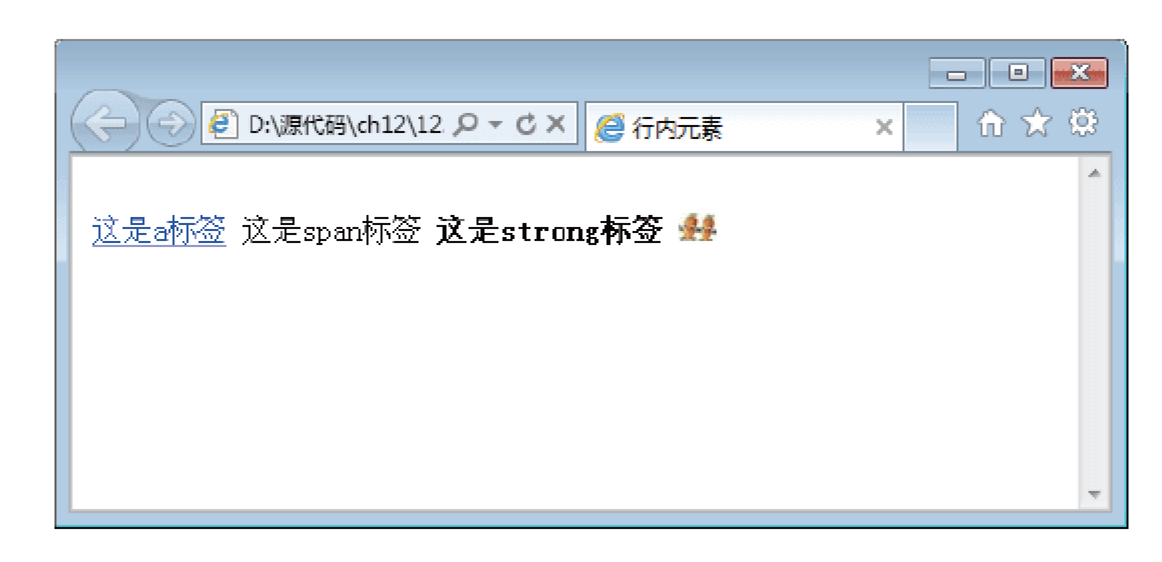


图12-10 行内显示



12.6 综合实例——定位布局新闻

一个美观大方的页面，必然是一个布局合理的页面。左右布局是网页中比较常见的一种方式，即根据信息种类不同，将信息在当前页面左右侧分别显示。本实例将利用前面学习的知识，创建一个左右布局的新闻页面，具体步骤如下所示。

01 分析需求。

首先需要将整个页面分为左右两个模块，左模块放置一类信息，右模块放置一类信息，可以设定其宽度和高度。实例完成后，效果如图12-11所示。



图12-11　左右显示信息

02 创建HTML页面，实现基本列表。

创建HTML页面，同时用DIV在页面中划分左边DIV层和右边DIV层两个区域，并且将信息放入到相应的DIV层中，注意DIV层内引用CSS样式名称。

```
<html>
<head>
<title>布局</title>
</head>
<body>
<center>
<div class="big">
  <p class=pp>女人</p>
  <div class="left">
    <h1>女人</h1>
    <p> ·男人性福告白：女人的性感与年龄成正比09:59 </p>
    <p> ·六类食物能有效对抗紫外线11:15 </p>
    <p> ·打造夏美人 受OL追捧的清爽发型10:05 </p>
    <p> ·美丽帮帮忙：别让大油脸吓跑男人09:47 </p>
    <p> ·简约雪纺清凉衫 百元搭出欧美范儿14:51 </p>
    <p> ·花边连衣裙超勾人 7月穿搭出新意11:04 </p>
  </div>
  <div class="right">
    <h1>健康</h1>
```

```
        <p> ·女性养生：让女人老得快的10个原因19:18 </p>
        <p> ·养生盘点：喝豆浆的九大好处和七大禁忌09:14</p>
        <p> ·养生警惕：14个护肤心理“错”觉19:57</p>
        <p> ·柿子番茄骨汤 8种营养师最爱的食物15:16</p>
        <p> ·夏季养生指南：“夫妻菜”宜常吃10:48 </p>
        <p> ·10条食疗养生方法，居家宅人的养生经13:54 </p>
      </div>
    </div>
    </center>
    </body>
    </html>
```

在IE9.0中浏览效果如图12-12所示，可以看到页面显示了两个模块，分别是“女人”和“健康”，二者上下排列。

03 添加CSS代码，修饰整体样式和DIV层。

```
<style>
* {
    padding:0px;
    margin:0px;
}body {
    font:"宋体";
    font-size:12px;
}
.big{
    width:570px;
                height:210px;
                border:#C1C4CD 1px solid;
    }

</style>
```

在IE9.0中浏览效果如图12-13所示，可以看到页面字体比原来的小，并且大的DIV显示了边框。

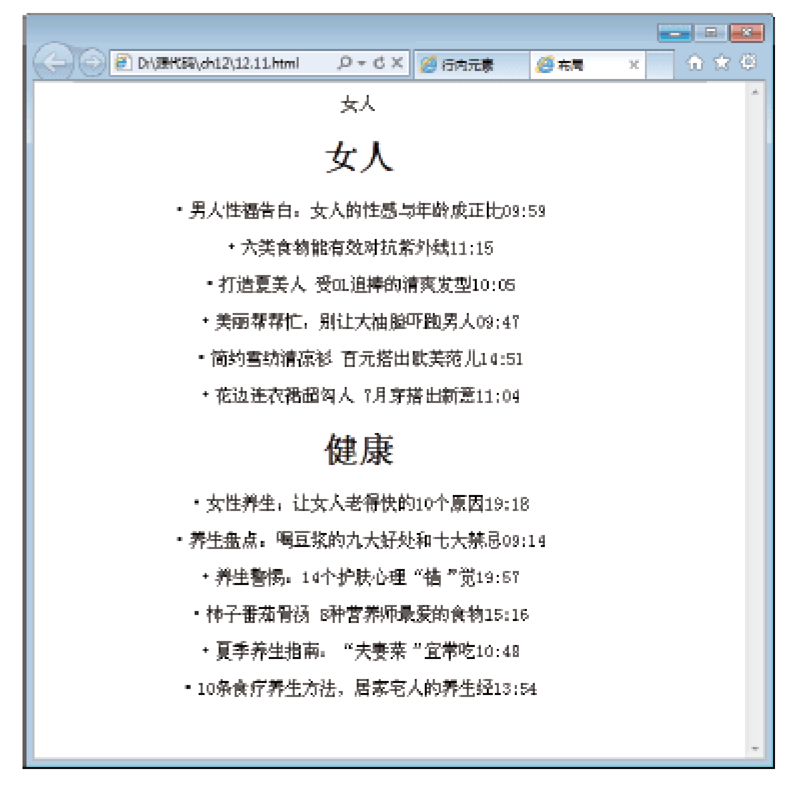


图12-12 上下排列

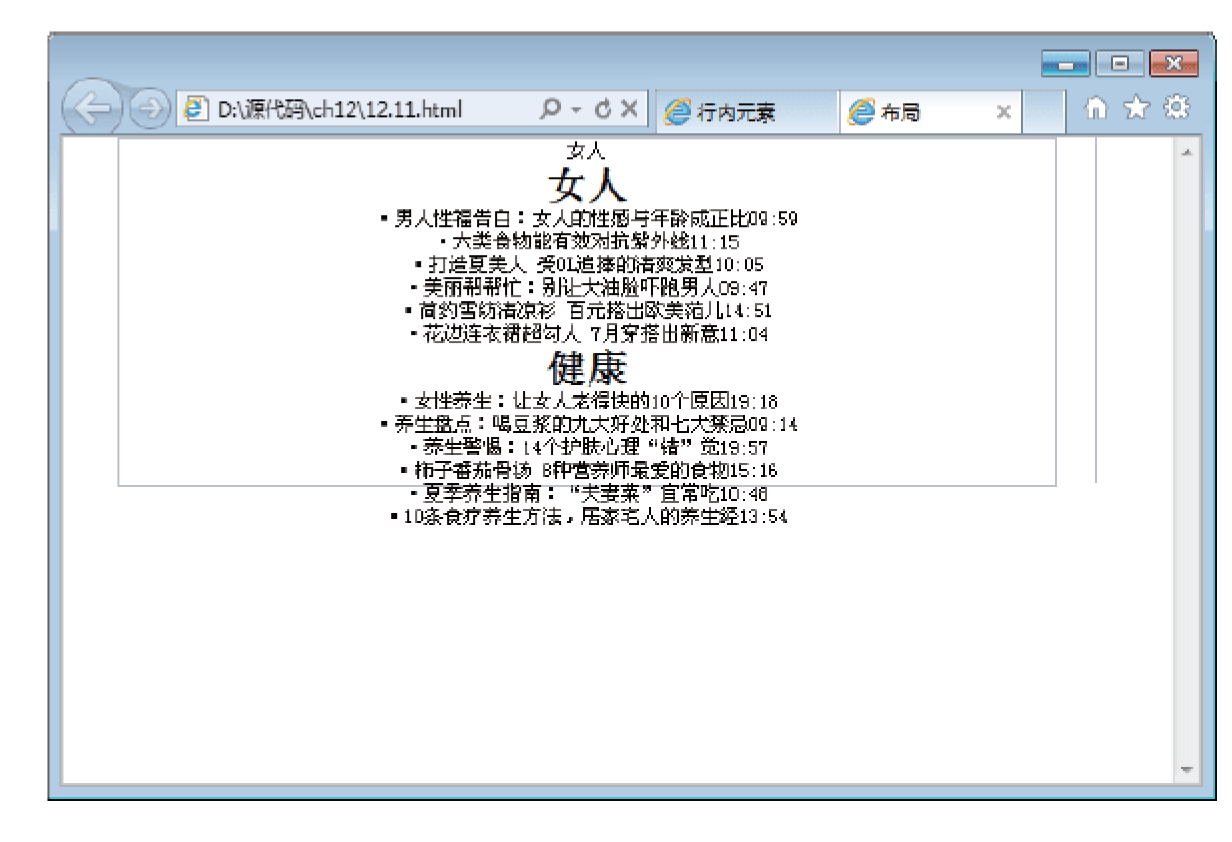


图12-13 修饰整体样式

04 添加CSS代码，设置两个层左右并列显示。

```
.left{
    width:280px;
    float:right; //设置右边悬浮
    border:#C1C4CD 1px solid;
    }.right{
    width:280px;
    float:left;//设置左边悬浮
    margin-left:6px;
    border:#C1C4CD 1px solid;
    }
```

在IE9.0中浏览效果如图12-14所示，可以看到页面中文本信息左右并列显示，但字体没有发生变化。

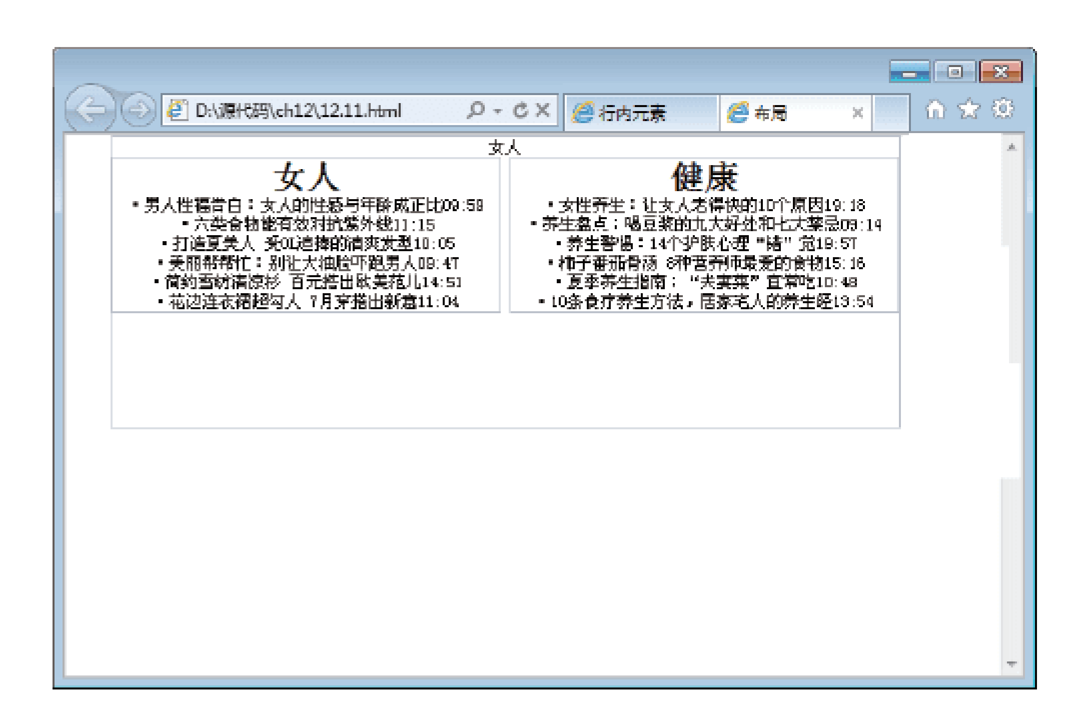


图12-14 设置左右悬浮

05 添加CSS代码，定义文本样式。

```
h1{
    font-size:14px;
    padding-left:10px;
    background-color:#CCCCCC;
    height:20px;
    line-height:20px;
    }p{
    margin:5px;
    line-height:18px;
                color:#2F17CD;
    }.pp{
                width:570px;
                text-align:left;
                height:20px;
                background-color:D5E7FD;
                position:relative;
                left:-3px;
                top:-3px;
                font-size:16px;
                text-decoration:underline;
}
```

在IE9.0中浏览效果如图12-11所示，可以看到页面中文本信息左右并列显示，其字体颜色为蓝色，行高为18px。



12.7 专家解惑

1．块元素和行内元素的概念分别是什么？有哪些特点？

块级元素和行内元素是布局最基本的两种元素，常见块级元素有div、p、form、ul、ol、li等，常见行内元素有span、strong、em等。

块级元素会独占一行，对应于display:block，可以设置width、height、margin、padding 属性；行内元素不独占一行，对应于display:inline，相邻行内元素会排列在同一行里，直到排不下才换行。设置width、height属性无效，而margin和padding属性只对设置水平方向的right和left有效，可以通过修改display属性来切换块级元素和行内元素。

display:inline-block是行内的块级元素，拥有块级元素的特点，可以设置 width、height、margin、padding 值，但又可以和其他行内元素排在同一行里。

2．当设置多个div并列时，为什么会撑破整个布局？

很多时候，尤其是容器内有平行布局（如两三个float的div）时，宽度很容易出现问题。在IE中，外层的宽度会被内层更宽的div挤破。一定要用Photoshop或者Firework量取像素级的精度。



CSS3盒子和DIV布局

本章引言

在传统网页设计中，为了设定页面元素的位置，经常使用表格来完成。表格起到了定位和布局的作用。由于表格布局的局限性，目前往往采用DIV+CSS3方式。CSS3中提出了盒子模型和新增盒子模型来完成对元素的直接定位，即能够为页面元素定义边框，并修饰内容距离，从而优化文本内容的显示效果。本章节将学习CSS3盒子和DIV布局的方法和技巧。







13.1 认识div层

使用DIV进行网页排版，是现在流行的一种趋势。例如，使用CSS3属性可以轻易设置DIV位置，演变出多种不同的布局方式。

13.1.1 层在HTML布局应用

<div>标记是一个区块容器标记，在<div></div>标记中可以放置其他一些HTML元素，例如段落<p>、标题<h1>、表格<table>、图片<img>和表单等。然后使用CSS3相关属性将div容器标记中的元素作为一个独立对象进行修饰。这样就不会影响其他HTML元素。

【例13.1】实例文件：ch13\13.1.html

```
<html>
<head>
<title>div 层</title>
<style type="text/css">
<!--
div{
  font-size:18px;
  font-weight:bolder;
  font-family:"幼圆";
  color:#FF0000;
  background-color:#eeddcc;
  text-align:center;
  width:300px;
  height:100px;
              border:1px #992211 dotted;
}
-->
</style>
   </head>
<body>
<center>
  <div>
  这是div层
  </div>
</center>
</body>
</html>
```

上面例子通过CSS对div块进行控制，绘制了一个div容器，容器中放置了一段文字。

在IE9.0中浏览效果如图13-1所示，可以看到一个矩形方块的div层，居中显示，字体显示为红色，边框为浅红色，背景色为浅黄色。

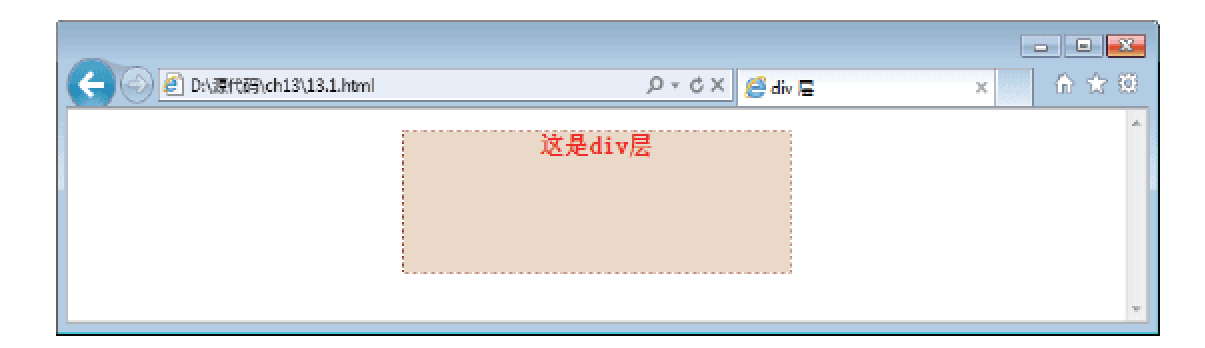


图13-1 div层显示

13.1.2 div和span区别

对于初学者而言，div和span两个标记常常被混淆，因为大部分的div层都可以使用span标记代替，并且其运行效果完全一样，例如13.1.1节中的例子，就可以使用span标记代替div标记，其运行效果完全一致。可以这样说，在使用区块对HTML元素进行包含方面，div和span标记作用基本一样。

div和span标记二者的区别在于，div是一个块级元素，其包含的元素会自动换行；span标记是一个行内标记，其前后都不会发生换行。div标记可以包含span标记元素，但span标记一般不包含div标记。

【例13.2】实例文件：ch13\13.2.html

```
<html>
<head>
<title>div与span的区别</title>
   </head>
<body>
  <p>div自动分行：</p>
  <div><b>宁静</b></div>
  <div><b>致远</b></div>
  <div><b>明治</b></div>
  <p>span同一行：</p>
  <span><b>老虎</b></span>
  <span><b>狮子</b></span>
  <span><b>老鼠</b></span>
</body>
</html>
```

在IE9.0中浏览效果如图13-2所示，可以看到div层所包含的元素，进行自动换行，而对于span标记，3个HTML元素在同一行显示。

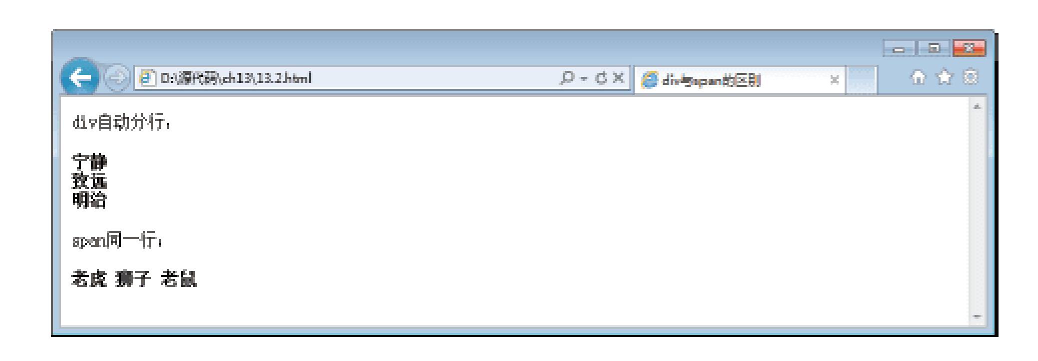


图13-2 div和span比较



在网页设计中，对于较大的块可以使用div完成，而对于具有独特样式的单独HTML元素，可以使用span标记完成。

13.2 盒子模型

将网页上每个HTML元素视为长方形的盒子，是网页设计上的一大创新。在控制页面方面，盒子模型有着至关重要的作用，熟练掌握盒子模型及其各个属性，是控制页面中每个HTML元素的前提。

13.2.1 什么是盒子模型

CSS3中，所有的页面元素都包含在一个矩形框内，这个矩形框就称为盒子。盒子描述了元素及其属性在页面布局中所占的空间大小，因此盒子可以影响其他元素的位置及大小。例如，页面中第一个盒子为10px，那么下一个盒子就处于离顶部10px距离的位置。如果第一个盒子增加为20px，则下一个盒子就要再下移10px。而整个页面就是由这些大大小小但不会重叠的盒子形成的。

盒子模型是由margin（边界）、border（边框）、padding（空白）和content（内容）几个属性组成。此外在盒子模型中，还具备高度和宽度两个辅助属性。盒子模型如图13-3所示。

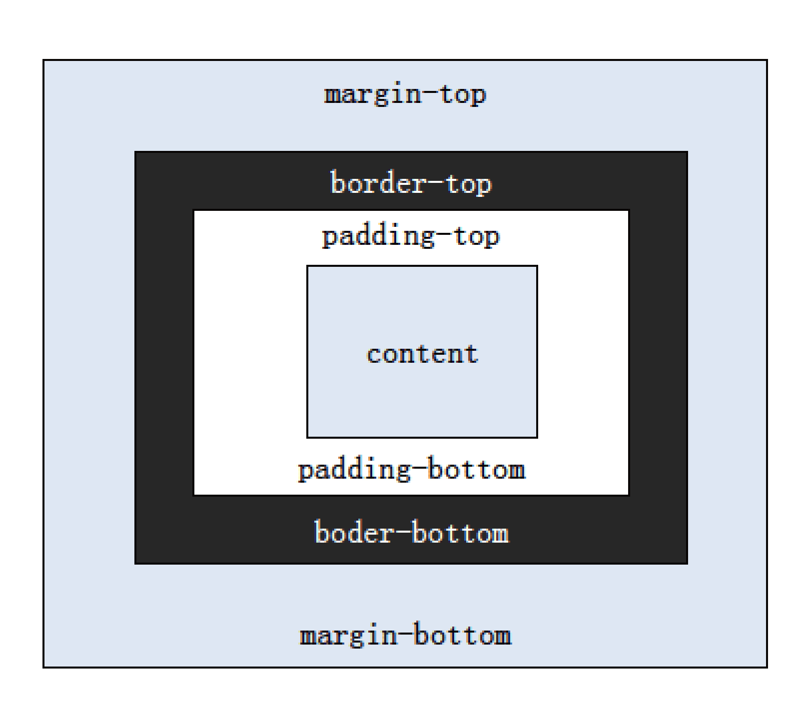


图13-3 盒子模型效果

从图13-3中可以看出，盒子模型包含如下4个部分：

- content：是盒子模型中必需的一部分，可以是文字、图片等元素。
- padding：也称内边距或补白，用来设置内容和边框之间的距离。
- border：可以设置内容边框线的粗细、颜色和样式等，前面已经接触过。
- margin：外边距，用来设置内容与内容之间的距离。

对于这些属性可以把它转移到日常生活中的盒子（箱子）上来理解，日常生活中所见的盒子也就是能装东西的一种箱子，也具有这些属性，所以称其为盒子模式。content就是盒子里装的东西；padding就是怕盒子里装的东西（贵重的）损坏而添加的泡沫或者其他抗震的辅料；border就是盒子本身了；margin则说明盒子摆放的时候不能全部堆在一起，要留一定空隙保持通风，同时也为了方便取出。

提示 在网页设计上，内容常指文字、图片等元素，但是也可以是小盒子（DIV嵌套），与现实生活中盒子不同的是，现实生活中的东西一般不能大于盒子，否则盒子会被撑坏的，而CSS盒子具有弹性，里面的东西大于盒子本身时它会被撑大，但不会被损坏。

一个盒子的实际高度（宽度）是由content+padding+border+margin组成的。在CSS3中，可以通过设定width和height来控制content的大小，并且对于任何一个盒子，都可以分别设定4条边的border、padding和margin。

13.2.2 border边框

border边框是内边距和外边距的分界线，可以分离不同的HTML元素，border的外围是元素的最外围。在网页设计中，如果计算元素的宽和高则需要把border计算在内。

border有3个属性，分别是边框样式（style）、颜色（color）和宽度（width），这些内容在第7章已经介绍过，这里就不再重复了。

【例13.3】实例文件：ch13\13.3.html

```
<html>
<head>
<title>border边框</title>
  <style type="text/css">
     .div1{
      borde-widthr:10px;
     border-color:#ddccee;
     border-style:solid;
     width:410px;
     }
     .div2{
      border-width:1px;
     border-color:#adccdd;
     border-style:dotted;
     width:410px;
     }
    .div3{
      border-width:1px;
     border-color:#457873;
     border-style:dashed;
     width:410px;
     }
  </style>
</head>
<body>
  <div class="div1">
     这是一个宽度为10px的实线边框。
    </div>
    <br /><br />
    <div class="div2">
     这是一个宽度为1px的虚线边框。
```

```
        </div>
        <br /><br />
        <div class="div3">
          这是一个宽度为1px的点状边框。
        </div>
    </body>
    </html>
```

在IE9.0中浏览效果如图13-4所示，可以看到显示了3个不同风格的盒子，第一个盒子边框线宽度为10px，边框样式为实线，颜色为紫色；第二个盒子边框线宽度为1px，边框样式为虚线，颜色为浅绿色；第三个盒子边框宽度为1px，边框样式是点状，颜色为绿色。

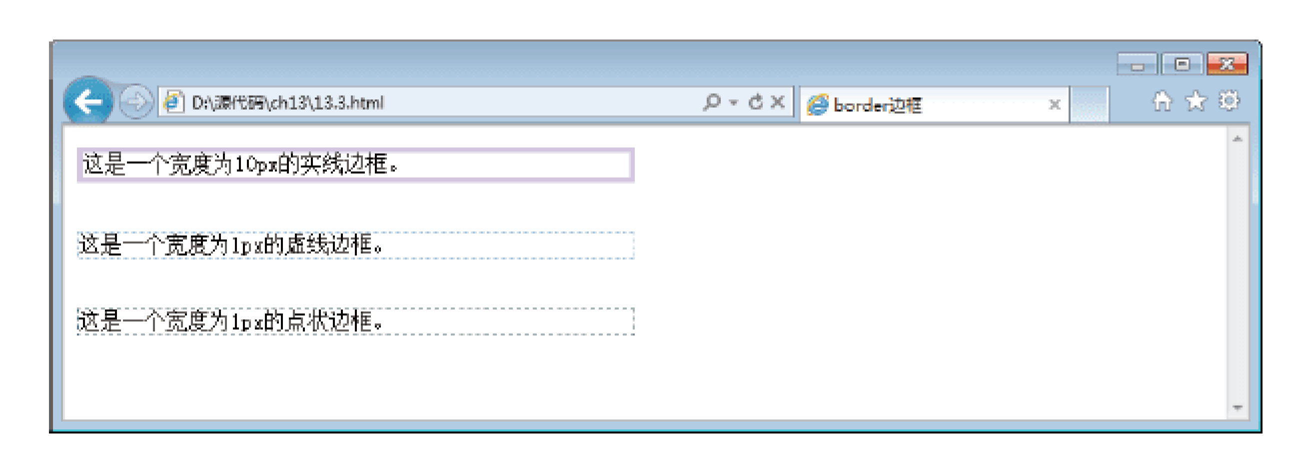


图13-4 设置盒子边框



这里需要注意的是，在给元素设置background-color背景色时，IE作用的区域为content+padding，而Firefox则是content+padding+border。这点在border为粗虚线时特别明显。

13.2.3 padding内边距

在CSS3中，可以设置padding属性定义内容与边框之间的距离，即内边距。

其语法格式如下：

```
padding : length
```

padding属性值可以是一个具体的长度，也可以是一个相对于上级元素的百分比，但不可以使用负值。当设置值为百分数时，百分数值是相对于其父元素的width计算的，这一点与外边距一样。所以，如果父元素的width改变，其值也会改变。

padding属性能为盒子定义上、下、左、右间隙的宽度，也可以单独定义各方位的宽度。常用形式如下：

```
padding :padding-top | padding-right | padding-bottom | padding-left
```

如果提供4个参数值，将按顺时针的顺序作用于4条边。如果只提供1个参数值，将作用于全部的4条边；如果提供2个，第1个作用于上下两边，第2个作用于左右两边，如果提供3个，第1个作用于上边，第2个作用于左、右两边，第3个作用于下边。

padding子属性及其含义如表13-1所示。

表13-1 padding子属性

属性	描述
padding-top	设定上间隙
padding-bottom	设定下间隙
padding-left	设定左间隙
padding-right	设定右间隙

【例13.4】实例文件：ch13\13.4.html

```
<html>
<head>
<title>padding</title>
  <style type="text/css">
    .wai{
      width:400px;
      height:250px;
      border:1px #993399 solid;
    }
    img{
         max-height:120px;
      padding-left:50px;
      padding-top:20px;
      }
  </style>
</head>
<body>
  <div class="wai">
    <img src="13.jpg" />
        <p>这张图片的左内边距是50px，顶内边距是20px</p>
    </div>
</body>
</html>
```

在Firefox 5.0中浏览效果如图13-5所示，可以看到一个div层中，显示了一个图片。此图片可以看做一个盒子模型，并定义了图片的左内边距和上内边距的效果。可以看出，内边距其实是对象img和外层div之间的距离。



图13-5 设置内边距

13.2.4 margin外边距

margin边界用来设置页面中元素和元素之间的距离，即定义元素周围的空间范围，是页面排版中一个比较重要的概念。

其语法格式如下：

```
margin : auto | length
```

其中，auto表示根据内容自动调整，length表示由浮点数字和单位标识符组成的长度值或百分数。百分数是基于父对象的高度。对于内联对象来说，左右外延边距可以是负数值。

margin属性包含的4个子属性控制一个页面元素的四周的边距样式，如表13-2所示。

表13-2 margin子属性

属性	描述
margin-top	设定上边距
margin-bottom	设定下边距
margin-left	设定左边距
margin-right	设定右边距

各子属性的属性值同样可以是一个确定的长度，也可以是一个百分比，该百分比是相对于其上级元素的宽度（width）。

在给margin设置值时，如果提供4个参数值，将按顺时针的顺序作用于4条边；如果只提供1个参数值，将作用于全部的4条边；如果提供2个，第1个作用于上下两边，第2个作用于左右两边；如果提供3个，第1个作用于上边，第2个作用于左、右两边，第3个作用于下边。

如果希望很精确地控制块的位置，需要对margin有更深入的了解。margin设置可以分为行内元素块之间设置、非行内元素块之间设置和父子块之间设置。

1. 行内元素margin设置

【例13.5】实例文件：ch13\13.5.html

```
<html>
<head>
<title>行内元素设置margin</title>
<style type="text/css">
<!--
span{
  background-color:#a2d2ff;
  text-align:center;
  font-family:"幼圆";
  font-size:12px;
  padding:10px;
              border:1px #ddeecc solid;
}
span.left{
  margin-right:20px;
  background-color:#a9d6ff;
```

```
}
span.right{
  margin-left:20px;
  background-color:#eeb0b0;
}
-->
</style>
   </head>
<body>
  <span class="left">行内元素1</span><span class="right">行内元素2</span>
</body>
</html>
```

在IE9.0中浏览效果如图13-6所示，可以看到一个蓝色盒子和红色盒子，两者之间的距离使用margin设置，其距离是左边盒子的右边距margin-right加上右边盒子的左边距margin-left。

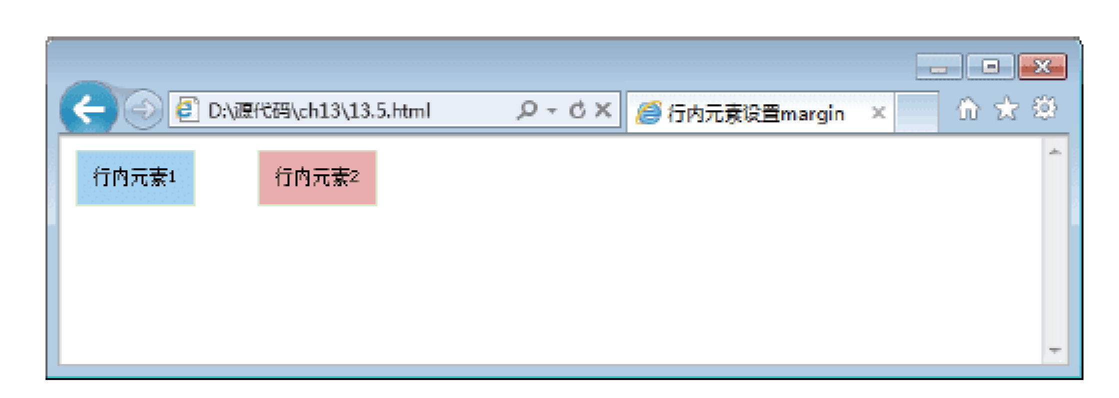


图13-6 设置外边距

2．非行内元素块之间margin设置

如果不是行内元素，而是产生换行效果的块级元素，情况就会发生变化。两个换行块级元素之间的距离不再是margin-bottom和margin-top的和，而是两者中的较大者。

【例13.6】实例文件：ch13\13.6.html

```
<html>
<head>
<title>块级元素的margin</title>
<style type="text/css">
<!--
h1{
  background-color:#ddeecc;
  text-align:center;
  font-family:"幼圆";
  font-size:12px;
  padding:10px;
              border:1px #445566 solid;
              display:block;
}
-->
</style>
   </head>
<body>
  <h1 style="margin-bottom:50px;">距离下面块的距离</h1>
  <h1 style="margin-top:30px;">距离上面块的距离</h1>
```

```
</body>
</html>
```

在IE9.0中浏览效果如图13-7所示，可以看到两个h1盒子，两者上下之间存在距离，其距离为margin-bottom和margin-top中较大的值，即50px。如果修改下面h1盒子元素的margin-top为40px，会发现执行结果没有任何变化。如果其值修改为60px，会发现下面的盒子会向下移动10px。



图13-7 设置上下margin距离

3. 父子块之间margin设置

当一个div块包含在另一个div块中间时，二者便会形成一个典型的父子关系。其中子块的margin设置将会以父块的content为参考。

【例13.7】实例文件：ch13\13.7.html

```
<html>
<head>
<title>包含块的margin</title>
<style type="text/css">
<!--
div{
  background-color:#fffebb;
  padding:10px;
  border:1px solid #000000;
}
h1{
  background-color:#a2d2ff;
  margin-top:0px;
  margin-bottom:30px;
  padding:15px;
  border:1px dashed #004993;
              text-align:center;
  font-family:"幼圆";
  font-size:12px;
}
-->
</style>
   </head>
<body>
  <div >
```

```
        <h1>子块div</h1>
      </div>
    </body>
    </html>
```

在IE9.0中浏览效果如图13-8所示，可以看到子块h1盒子距离父div下边界为40px（子块30px的外边距加上父块10px的内边距），其他3边距离都是父块的padding距离，即10px。

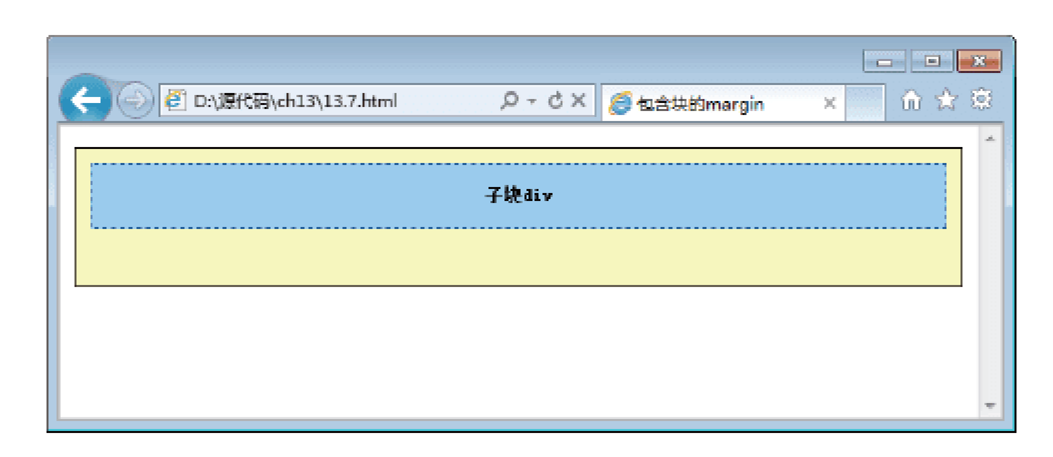


图13-8 设置包括盒子的margin距离

在上例中，如果设定了父元素的高度height值，并且父块高度值小于子块的高度加上margin的值，此时IE浏览器会自动扩大，保持子元素的margin-bottom的空间以及父元素的padding-bottom。而Firefox就不会这样，会保证父元素的height高度的完全吻合，而这时子元素将超过父元素的范围。

当将margin设置为负数时，会使得被设为负数的块向相反的方向移动，甚至覆盖在另外的块上。



13.3 CSS3新增弹性盒子模型

CSS3引入了新的盒子模型处理机制，即弹性盒子模型。该模型决定元素在盒子中的分布方式以及如何处理盒子的可用空间。通过弹性盒子模型，可以轻松地设计出自适应浏览器窗口的流动布局或自适应字体大小的弹性布局。

CSS3为弹性盒子模型新增了8个属性，如表13-3所示。

表13-3 CSS3新增盒子模型属性

属性	说明
box-orient	定义盒子分布的坐标轴
box-align	定义子元素在盒子内垂直方向上的空间分配方式
box-direction	定义盒子的显示顺序
box-flex	定义子元素在盒子内的自适应尺寸
box-flex-group	定义自适应子元素群组
box-lines	定义子元素分布显示
box-ordinal-group	定义子元素在盒子内的显示位置
box-pack	定义子元素在盒子内的水平方向上的空间分配方式

13.3.1 盒子布局取向box-orient

box-orient属性用于定义盒子元素内部的流动布局方向，即横排还是竖排。

其语法格式如下：

```
box-orient:horizontal | vertical | inline-axis | block-axis | inherit
```

其属性值含义如表13-4所示。

表13-4 box-orient属性值

属性值	说明
horizontal	盒子元素从左到右在一条水平线上显示它的子元素
vertical	盒子元素从上到下在一条垂直线上显示它的子元素
inline-axis	盒子元素沿着内联轴显示它的子元素
block-axis	盒子元素沿着块轴显示它的子元素

弹性盒子模型是W3C标准化组织于2009年发布的，目前还没有主流浏览器对其支持，不过采用Webkit和Mozilla渲染引擎的浏览器都自定义了一套私有属性，用来支持弹性盒子模型。下面的代码中会存在一些Firefox浏览器的私有属性定义。

【例13.8】实例文件：ch13\13.8.html

```
<html>
<head>
<title>box-orient</title>
<style>
div{height:50px;text-align:center;}
.d1{background-color:#F6F;width:180px;height:500px}
.d2{background-color:#3F9;width:600px;height:500px}
.d3{background-color:#FCd;width:180px;height:500px}
body{
       display:box;/*标准声明，盒子显示*/
        display:-moz-box;/*兼容Mozilla Gecko引擎浏览器*/
        orient:horizontal;/*定义元素为盒子显示*/
        -mozbox-box-orient:horizontal;/*兼容Mozilla Gecko引擎浏览器*/
       box-orient:horizontal;/*CSS3标准化设置*/
}
</style>
</head>
<body>
<div class=d1>左侧布局</div>
<div class=d2>中间布局</div>
<div class=d3>右侧布局</div>
</body>
</html>
```

上面代码中，CSS样式首先定义了每个div层的背景色和大小，在body标记选择器中，定义了body容器中元素以盒子模型显示，并使用box-orient定义元素水平并列显示。

在Firefox 5.0中浏览效果如图13-9所示，可以看到显示了3个层，3个div层并列显示，分

别为“左侧布局”、“中间布局”和“右侧布局”。



图13-9 盒子元素水平并列显示

13.3.2 盒子布局顺序box-direction

box-direction是用来确定子元素的排列顺序，也可以说是内部元素的流动顺序。

其语法格式如下：

```
box-direction:normal | reverse | inherit
```

其属性值如表13-5所示。

表13-5 box-direction属性值

属性值	说明
normal	正常显示顺序，即如果盒子元素的box-orient属性值为horizontal，则其包含的子元素按照从左到右的顺序显示，也即每个子元素的左边总是靠近前一个子元素的右边；如果盒子元素的box-orient属性值为vertical，则其包含的子元素按照从上到下的顺序显示
reverse	反向显示，盒子所包含的子元素的显示顺序将与normal相反
inherit	继承上级元素的显示顺序

【例13.9】实例文件：ch13\13.9.html

```
<html>
<head>
<title>box-direction</title>
<style>
div{height:50px;text-align:center;}
.d1{background-color:#F6F;width:180px;height:500px}
.d2{background-color:#3F9;width:600px;height:500px}
.d3{background-color:#FCd;width:180px;height:500px}
body{
       display:box;/*标准声明，盒子显示*/
        display:-moz-box;/*兼容Mozilla Gecko引擎浏览器*/
        orient:horizontal;/*定义元素为盒子显示*/
        -mozbox-box-orient:horizontal;/*兼容Mozilla Gecko引擎浏览器*/
        box-orient:horizontal;/*css3标准声明*/
```

```
            -moz-box-direction:reverse;
            box-direction:reverse;

    }
    </style>
    </head>
    <body>
    <div class=d1>左侧布局</div>
    <div class=d2>中间布局</div>
    <div class=d3>右侧布局</div>
    </body>
    </html>
```

可以发现此实例代码和上一个实例代码基本相同，只不过多了一个box-direction属性设置，此处设置布局进行反向显示。

在Firefox 5.0中浏览效果如图13-10所示，可以发现与上一个图形相比较，左侧布局和右侧布局进行了互换。



图13-10 盒子布局顺序设置

13.3.3 盒子布局位置box-ordinal-group

box-ordinal-group属性设置盒子中每个子元素在盒子中的具体位置。

其语法格式如下：

```
box-ordinal-group:<integer>
```

参数值integer是一个自然数，从1开始，用来设置子元素的位置序号。子元素将根据这个属性从小到大进行排列。在默认情况下，子元素将根据元素的位置进行排列。如果不知道box-ordinal-group属性值的子元素，则其序号都默认为1，并且序号相同的元素将按照它们在文档中加载的顺序进行排列。

【例13.10】实例文件：ch13\13.10.html

```
<html>
<head>
<title>box-ordinal-group</title>
<style>
```

```
    body{
        margin:0;
        padding:0;
        text-align:center
        background-color:#d9bfe8;
    }
    .box{
         margin:auto;
         text-align:center;
         width:988px;
         display:-moz-box;
         display:box;
         box-orient:vertical;
         -moz-box-orient:vertical;
    }
    .box1{
            -moz-box-ordinal-group:2;
            box-ordinal-group:2;
    }
    .box2{
            -moz-box-ordinal-group:3;
            box-ordinal-group:3;
    }
    .box3{
            -moz-box-ordinal-group:1;
            box-ordinal-group:1;
    }
    .box4{
            -moz-box-ordinal-group:4;
            box-ordinal-group:4;
    }
    </style>
    </head>
    <body>
    <div class=box>
    <div class=box1><img src=1.jpg/></div>
    <div class=box2><img src=2.jpg/></div>
    <div class=box3><img src=3.jpg/></div>
    <div class=box4><img src=4.jpg/></div>
    </div>
    </body>
    </html>
```

在上面的样式代码中，类选择器box中代码display:box设置了容器以盒子模型显示，box-orient:vertical代码设置排列方向从上到下。在下面的box1、box2、box3和box4类选择器中通过box-ordinal-group属性都设置了显示顺序。

在Firefox 5.0中浏览效果如图13-11所示，可以看到第三个层次显示在第一个和第二个层次之上。

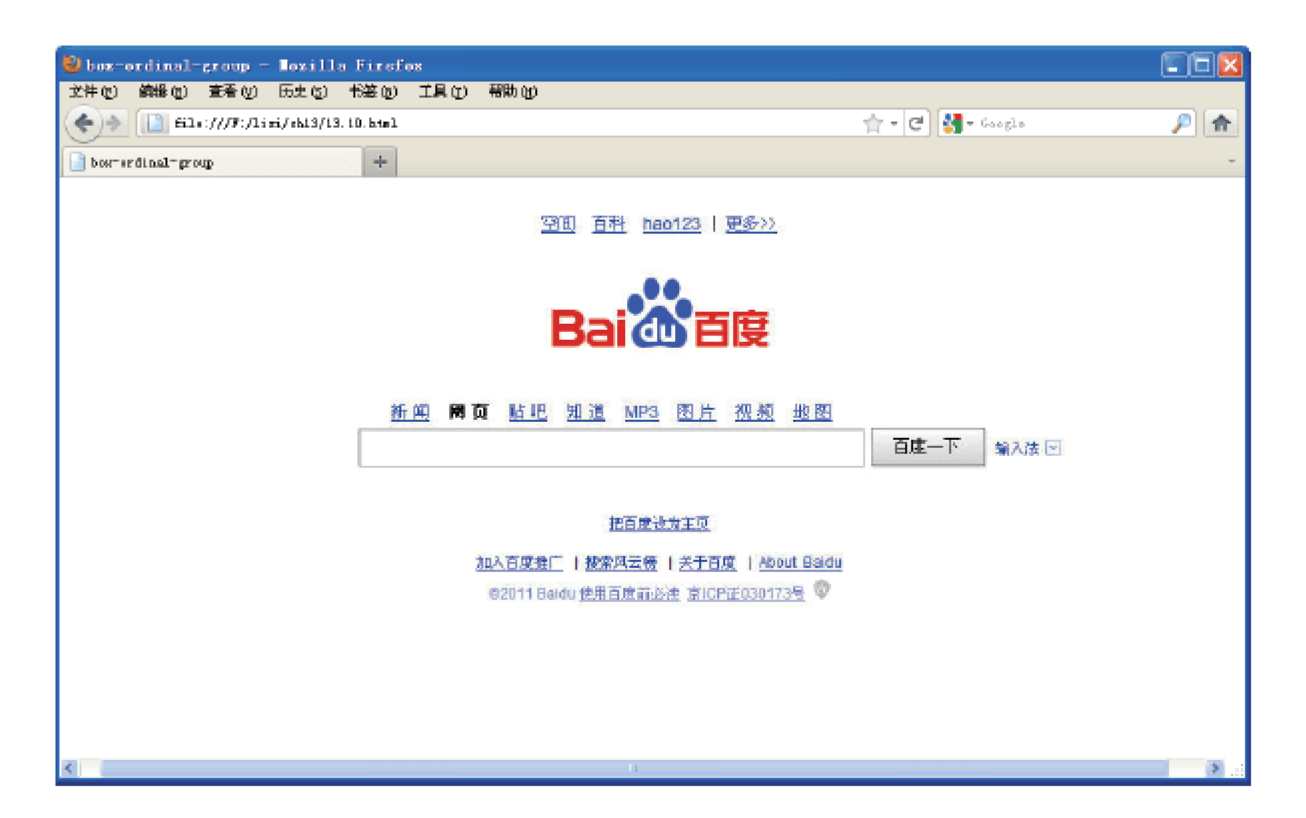


图13-11 设置层显示顺序

13.3.4 盒子弹性空间box-flex

box-flex属性能够灵活地控制子元素在盒子中的显示空间。显示空间包括子元素的宽度和高度，而不只是子元素所在栏目的宽度，也可以说是子元素在盒子中所占的面积。

其语法格式如下：

```
box-flex:<number>
```

number属性值是一个整数或者小数。当盒子中包含多个定义了box-flex属性的子元素时，浏览器将会把这些子元素的box-flex属性值相加，然后根据它们各自的值占总值的比例来分配盒子剩余的空间。



提示 box-flex属性只有在盒子拥有确定的空间大小时才能够正确解析，即为盒子定义具体的width和height属性值。

【例13.11】实例文件：ch13\13.11.html

```
<html>
<head>
<title>box-flex</title>
<style>
body{
margin:0;
padding:0;
text-align:center;
}
.box{
height:50px;
text-align:center;
width:960px;
overflow:hidden;
       display:box;/*标准声明，盒子显示*/
        display:-moz-box;/*兼容Mozilla Gecko引擎浏览器*/
```

```
            orient:horizontal;/*定义元素为盒子显示*/
            -mozbox-box-orient:horizontal;/*兼容Mozilla Gecko引擎浏览器*/
            box-orient:horizontal;/*css3标准声明*/
    }
    .d1{
    background-color:#F6F;
    width:180px;
    height:500px;
    }
    .d2,.d3{
      border:solid 1px #CCC;
      margin:2px;
    }
    .d2{
    -moz-box-flex:2;
    box-flex:2;
    background-color:#3F9;
    height:500px;
    }
    .d3{
    -moz-box-flex:4;
    box-flex:4;
    background-color:#FCd;
    height:500px;
    }
    .d2 div,.d3 div{display:inline;}
    </style>
    </head>
    <body>
    <div class=box>
    <div class=d1>左侧布局</div>
    <div class=d2>中间布局</div>
    <div class=d3>右侧布局</div>
    </div>
    </body>
    </html>
```

上面CSS样式代码中，使用display:box语句设定容器内元素以盒子方式布局，box-orient:horizontal语句设定盒子之间在水平方向上并列显示，类选择器d1中使用width和height设定显示层的大小，而在d2和d3中，使用box-flex分别设定两个盒子显示面积。

在Firefox 5.0中浏览效果如图13-12所示，可以看到左侧布局所占空间比中间布局较大。

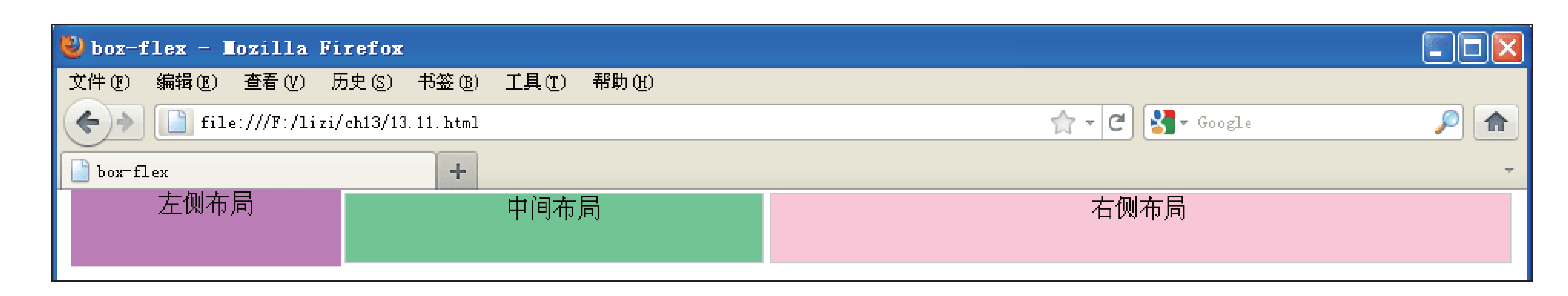


图13-12 设置盒子面积

13.3.5 管理盒子空间box-pack和box-align

当弹性元素和非弹性元素混合排版时，可能会出现所有子元素的尺寸大于或小于盒子的尺寸，从而出现盒子空间不足或者富余的情况，这时就需要一种方法来管理盒子的空间。如果子元素的总尺寸小于盒子的尺寸，则可以使用box-align和box-pack属性进行管理。

box-pack 属性可以用于设置子容器在水平轴上的空间分配方式，语法格式如下：

```
box-pack:start|end|center|justify
```

其属性值含义如表13-6所示。

表13-6 box-pack属性值

属性值	说明
start	所有子容器都分布在父容器的左侧，右侧留空
end	所有子容器都分布在父容器的右侧，左侧留空
justify	所有子容器平均分布（默认值）
center	平均分配父容器剩余的空间（能压缩子容器的大小，并且有全局居中的效果）

box-align 属性用于管理子容器在竖轴上的空间分配方式，语法格式如下：

```
box-align: start|end|center|baseline|stretch
```

其属性值含义如表13-7所示。

表13-7 box-align属性值

属性值	说明
start	子容器从父容器顶部开始排列，富余空间显示在盒子底部
end	子容器从父容器底部开始排列，富余空间显示在盒子顶部
center	子容器横向居中，富余空间在子容器两侧分配，上面一半下面一半
baseline	所有盒子沿着它们的基线排列，富余的空间可前可后显示
stretch	每个子元素的高度被调整到适合盒子的高度显示，即所有子容器和父容器保持同一高度

【例13.12】实例文件：ch13\13.12.html

```
<html>
<head>
<title>box-pack</title>
<style>
body,html{
height:100%;
width:100%;
}
body{
         margin:0;
         padding:0;
         display:box;/*标准声明，盒子显示*/
        display:-moz-box;/*兼容Mozilla Gecko引擎浏览器*/
        -mozbox-box-orient:horizontal;/*兼容Mozilla Gecko引擎浏览器*/
```

```
            box-orient:horizontal;/*css3标准声明*/
            -moz-box-pack:center;
            box-pack:center;
            -moz-box-align:center;
            box-align:center;
            background:#04082b url(a.jpg) no-repeat top center;
    }
    .box{
    border:solid 1px red;
    padding:4px;
    }
    </style>
    </head>
    <body>
    <div class=box>
    <img src=yueji.jpg>
    </div>
    </body>
    </html>
```

上面代码中，display:box语句定义容器内元素以盒子模型显示，box-orient:horizontal定义盒子水平显示，box-pack:center定义盒子两侧空间平均分配，box-align:center定义盒子上下两侧空间平均分配，即图片盒子居中显示。

在Firefox 5.0中浏览效果如图13-13所示，可以看到中间盒子在容器中部显示。

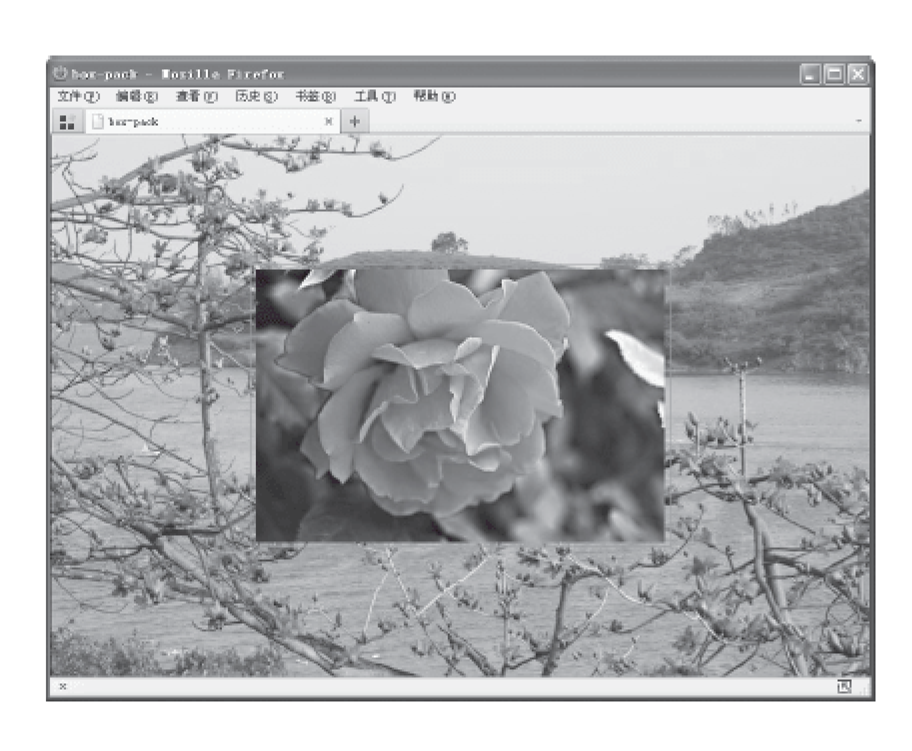


图13-13 设置盒子中间显示

13.3.6 空间溢出管理box-lines

弹性布局中盒子内的元素很容易出现空间溢出的现象，与传统的盒子模型一样，CSS3允许使用overflow属性来处理溢出内容的显示。当然还可以使用box-lines属性来避免空间溢出的问题。

其语法格式如下：

```
box-lines:single|multiple
```

其中，single表示子元素都单行或单列显示，multiple表示子元素可以多行或多列显示。

【例13.13】实例文件：ch13\13.13.html

```
<html>
<head>
<title>box-lines</title>
<style>
.box{
border:solid 1px red;
width:600px;
height:400px;
display:box;/*标准声明，盒子显示*/
display:-moz-box;/*兼容Mozilla Gecko引擎浏览器*/
-moz-box-orient:horizontal;/*兼容Mozilla Gecko引擎浏览器*/
-moz-box-lines:multiple;
box-lines:multiple;
}
.box div{
    margin:4px;
    border:solid 1px #aaa;
    -moz-box-flex:1;
    box-flex:1;
}
.box div img{width120px;}
</style>
</head>
<body>
<div class=box>
<div><img src="b.jpg"></div>
<div><img src="c.jpg"></div>
<div><img src="d.jpg"></div>
<div><img src="e.jpg"></div>
<div><img src="f.jpg"></div>
</div>
</body>
```

在Firefox 5.0中浏览效果如图13-14所示，可以看到右边盒子还是发生了溢出现象。这是因为目前各大主流浏览器还没有明确支持这种用法，所以导致box-lines属性被实际应用时显示无效。相信在未来的一段时间内，各个浏览器会支持该属性。

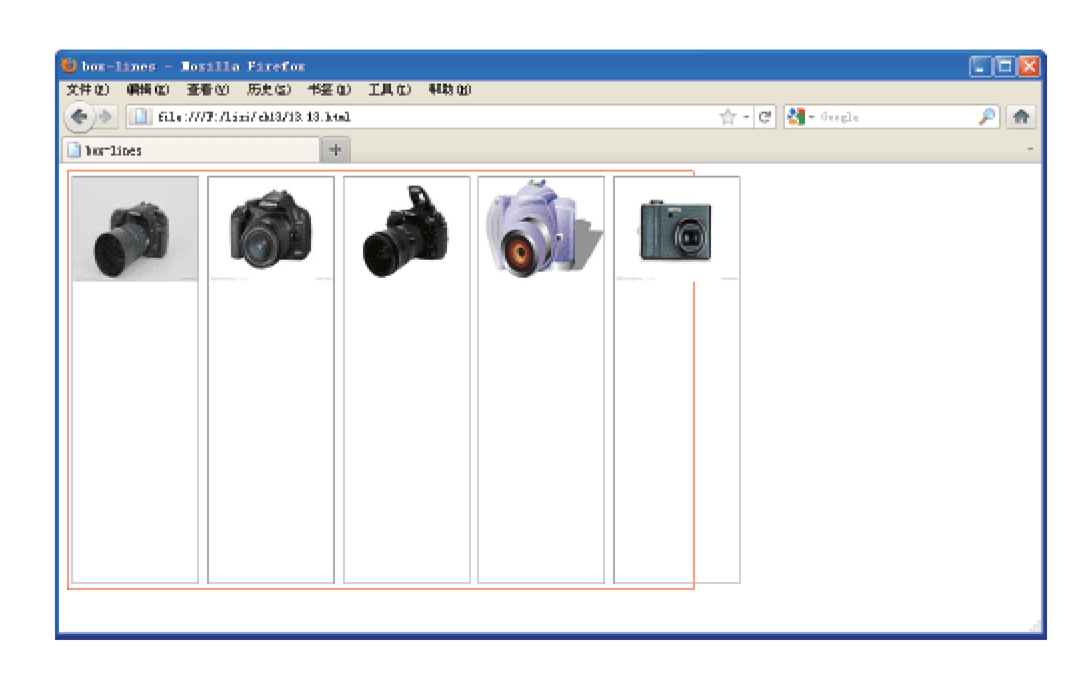


图13-14 溢出管理



13.4 综合实例1——图文排版效果

一个宣传页需要包括文字和图片信息。本实例将结合前面学习的盒子模型及其相关属性，创建一个旅游宣传页，具备步骤如下所示。

01 分析需求。

整个宣传页面需要一个div层包含并带有边框，DIV层包括两个部分，上部空间包含一个图片，下面显示文本信息并带有底边框，下部空间显示两张图片。实例完成后，效果如图13-15所示。

02 构建HTML页面，使用DIV搭建框架。

```
    <html>
    <head>
    <title>图文排版</title>
    </head>
    <body>
      <div class="big">
          <div class="up">
              <img src="top.jpg" border="0" />
                 <p>·反季游正流行 众信旅游暑期邀你到南半球过冬 </p>
                 <p> ·西安世园会暨旅游推介会今日在沈阳举行！</p>
                 <p> ·澳大利亚旅游局中国区首代邓李宝茵八月底卸任</p>
                 <p> ·“彩虹部落”土族:旅游经济支撑下的文化记忆恢复(组图)</p>
            </div>
            <div class="down">
              <img src="bottom1.jpg" border="0" />    <img
src="bottom2.jpg" border="0" />
            </div>
        </div>
    </body>
    </html>
```

在IE9.0中浏览效果如图13-16所示，可以看到页面自上向下，显示图片、段落信息和图片。

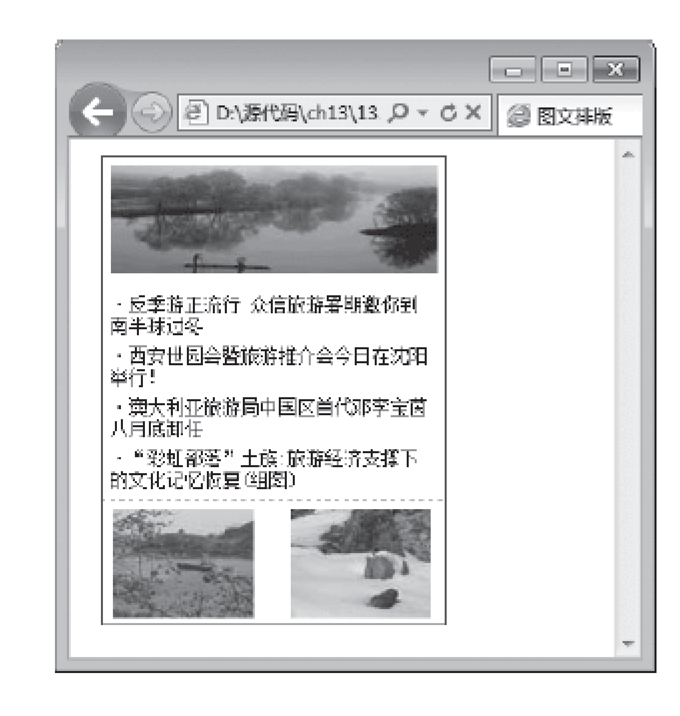


图13-15 旅游宣传页

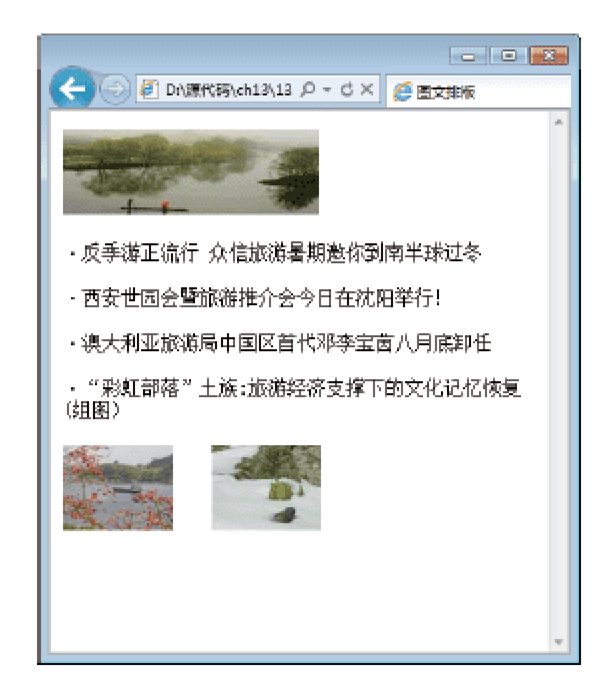


图13-16 构建HTML文档

03 添加CSS代码，修饰整体DIV。

```
<style>
  *{
    padding:0px;
    margin:0px;
    }
  body{
    font-family:"宋体";
    font-size:12px;
    }
  .big{
    width:220px;
    border:#0033FF 1px solid;
    margin:10px 0 0 20px;
    }
</style>
```

CSS样式代码在body标志选择器设置了字体和字体大小，并在big类选择器中，设置整个层的宽度、边框样式和外边距。

在IE9.0中浏览效果如图13-17所示，可以看到页面图片信息和文本都在一个矩形盒子内，其边框颜色为蓝色，大小为1px。

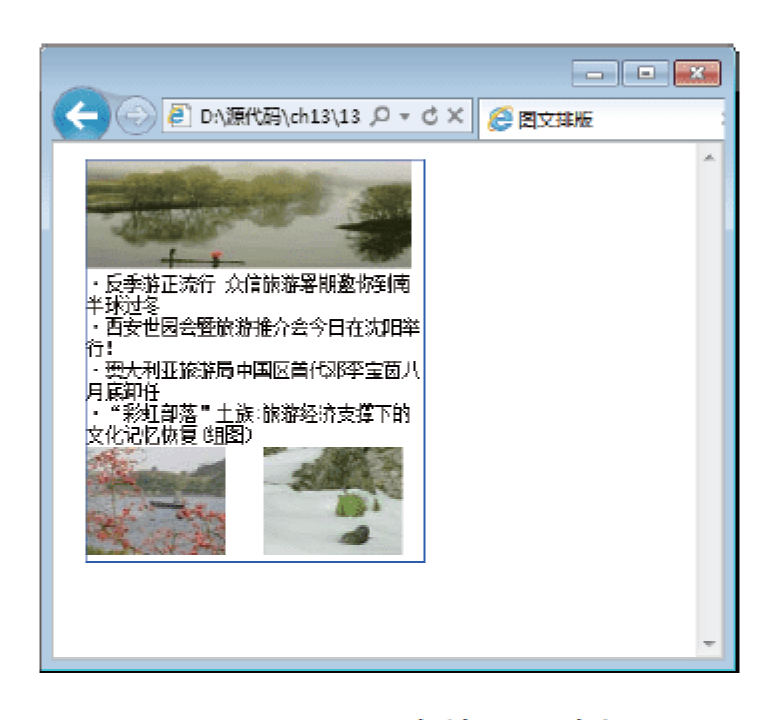


图13-17 设置整体DIV样式

04 添加CSS代码，修饰字体和图片。

```
  .up p{
    margin:5px;
    }
  .up img{
    margin:5px;
    text-align:center;}
  .down{
    text-align:center;
    border-top:#FF0000 1px dashed;
    }
  .down img{
    margin-top:5px;
    }
```

上面代码定义了段落、图片的外边距，例如margin-top:5px语句设置了下面图片的外边距为5px，两个图片距离是10px。

在IE9.0中浏览效果如图13-15所示，可以看到字体居中显示，下面带有一个红色虚线，宽度为1px。



13.5 综合实例2——淘宝导购菜单

网上购物已经成为了一种时尚，其中淘宝网是网上购物网站影响比较大的网站之一。淘宝网的宣传页面到处都是。本实例结合前面学习的知识，创建一个淘宝网宣传导航页面，具体步骤如下所示。

01 分析需求。

根据实际效果，需要创建一个div层包含3个部分，即左边导航栏、中间图片显示区域和右边导航栏，然后使用CSS颜色设置导航栏字体和边框。实例完成后，具体效果如图13-18所示。

02 构建HTML页面，使用DIV搭建框架。

```
<html >
<head>
<title>淘宝网</title>
</head>
<body>
<div class="wrap">
  <div class="area">
 <div >
    <div class="tab_area">
    <ul>
    <li class="current"><a href="#">男T恤</a></li>
    <li ><a href="#">男衬衫</a></li>
    <li><a href="#">休闲裤</a></li>
    <li><a href="#">牛仔裤</a></li>
                  <li><a href="#">男短裤</a></li>
                             <li><a href="#">西裤</a></li>
    <li><a href="#">皮鞋</a></li>
                             <li><a href="#">休闲鞋</a></li>
                             <li ><a href="#">男凉鞋</a></li>
                            </ul>
                        </div>
          <div class="tab_area1" >
  <ul>
<li><a href="#">女T恤</a></li>
   <li><a href="#">女衬衫</a></li>
   <li><a href="#">开衫</a></li>
   <li ><a href="#">女裤</a></li>
```

```
                        <li><a href="#">女包</a></li>
                        <li ><a href="#">男包</a></li>
         <li ><a href="#">皮带</a></li>
                        <li><a href="#">登山鞋</a></li>
                        <li ><a href="#">户外装</a></li>
        </ul>
        </div>
    </div>
    <div class="img_area">
        <img src=nantxu.jpg/>
    </div>
        </div>
    </body>
    </html>
```

在Firefox 5.0中浏览效果如图13-19所示，三部分内容分别自上而下显示，第一部分是导航菜单栏，第二部分也是一个导航菜单栏，第三部分是一个图片信息。



图13-18 淘宝宣传页

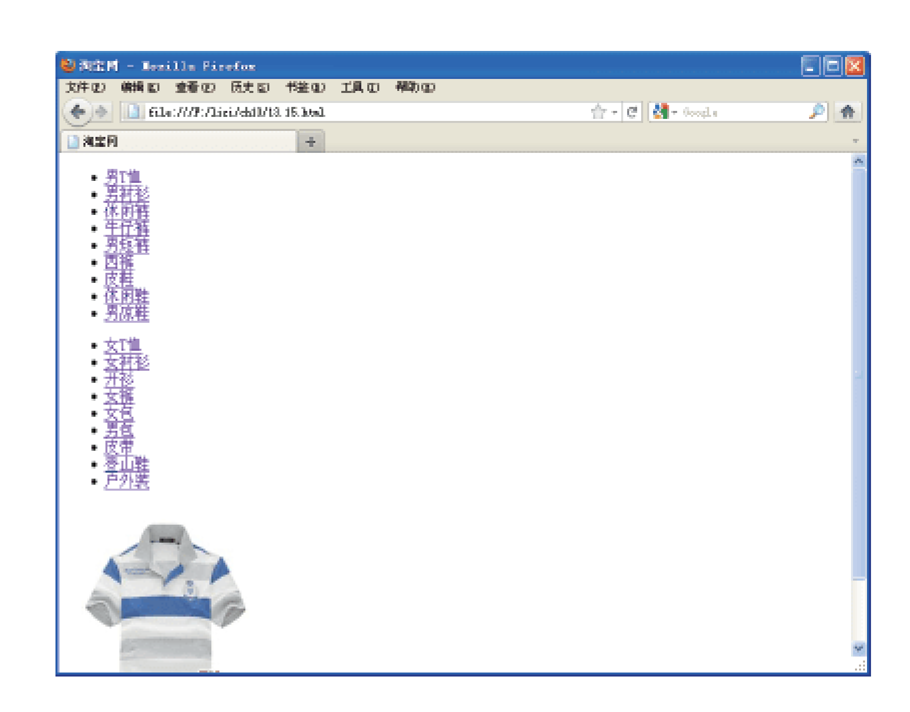


图13-19 基本HTML显示

03 添加CSS代码，修饰整体样式。

```
    <style type="text/css">
    body, p, ul, li{margin:0; padding:0;}
    body{font:12px arial,宋体,sans-serif;}
    .wrap{width:318px;height:248px; background-color:#FFFFFF;
float:left;border: 1px solid #F27B04;}
    .area{width:318px;  float:left;}
    .tab_area{width:53px; height:248px;  border-right:1px solid
#F27B04;overflow:hidden; }
    .tab_area1{width:53px; height:248px;  border-left:1px solid
#F27B04;overflow:hidden; position:absolute; left:265px; top:1px; }
    .img_area{
      width:208px;
      height:248px;
      overflow:hidden;
      position:absolute;
      top:-2px;
      left:55px;
    }
```

```
    </style>
```

上面CSS样式代码中，设置了body页面字体、段落、列表和列表选项的样式。需要注意的是类选择器tab_area定义了左边列表选项，即左边导航菜单，其宽度为53px，高度为248px，边框色为黄色。类选择tab_area1定义右边列表选项，即右边导航菜单，其宽度和高度同左侧菜单相同，但此次使用position定义了这个div层显示的绝对位置，语句为"position:absolute; left:265px; top:1px;"。类选择器img_area定义中间图片显示样式，也是使用position绝对定位。

在Firefox 5.0中浏览效果如图13-20所示，可以看到网页中显示了3个部分，左右两侧为导航菜单栏，中间是图片。



图13-20 设置整体布局样式

04 添加CSS代码，修饰列表选项。

```
    img{border:0;}
    li{list-style:none;}
    a{font-size:12px; text-decoration:none}
    a:link,a:visited {color:#999;}
    .tab_area ul li,.tab_area1 ul li
    {width:53px;height:27px;text-align:center;line-height:26px;
float:left;border-bottom:1px solid #F27B04;}
    .tab_area ul li a,.tab_area1 ul li a{color:#3d3d3d;}
    .tab_area ul li.current,.tab_area1 ul li.current{ height:27px; background-
color:#F27B04;}
    .tab_area ul li.current a,.tab_area1 ul li.current a{color:#fff; font-
size:12px; font-weight:400; line-height:27px}
```

上面CSS样式代码完成了对字体大小、颜色、是否带有下划线等属性定义。

在Firefox 5.0中浏览效果如图13-18所示，可以看到网页中左右两个导航菜单，相对于上面图形，字体颜色和大小发生变化。

13.6 专家解惑

1．如何理解margin的加倍问题？

当div层被设置为float时，在IE下设置的margin会加倍。其解决办法是，在这个div里面加上“display:inline;”。

例如：

```
<#div id="imfloat"></#div>
```

相应的css为：

```
#imfloat{
float:left;
margin:5px;
display:inline;
}
```

2．margin:0 auto表示什么含义？

margin:0 auto定义元素向上补白0像素，左右为自动使用。这样按照浏览器解析习惯是可以让页面居中显示的，一般这个语句会在body标记中。在使用margin:0 auto语句使页面居中的时候，一定要给元素一个高度并且不要让元素浮动，即不要加float，否则效果失效。



第 14 章

DIV+CSS3页面基本排版

本章引言

使用DIV+CSS3页面排版，已经逐步替代了使用Table表格排版，相比较Table表格，DIV+CSS3页面排版是一种更好的、更有亲和力的、更灵活的而且功能更强大的网站设计方法。







14.1 CSS3排版观念

DIV在CSS3+DIV页面排版中是一个块的概念，其起始标记和结束标记之间的所有内容都是用来构成这个块的，其中所包含元素特性由DIV标记属性来控制，或者通过使用样式表格式化这个块来进行控制。CSS3+DIV页面排版思想是，首先在整体上进行<div>标记的分块，然后对各个块进行CSS定位，最后再在各个块中添加相应的内容。

14.1.1 将页面用DIV分块

使用DIV+CSS3页面排版布局，需要对网页有一个整体构思，即网页可以划分几个部分，例如上中下结构、左右两列结构、三列结构。这时就可以根据网页构思，将页面划分为几个DIV块，用来存放不同的内容。当然了，大块中还可以存放不同的小块。最后，通过CSS属性，对这些DIV进行定位。

在现在的网页设计中，一般情况下的网站都是上中下结构，即上面是页面头部，中间是页面内容，最下面是页脚，整个上中下结构最后放到一个DIV容器中，方便控制。页面头部一般用来存放Logo和导航菜单，页面内容包含页面要展示的信息、链接和广告等，页脚存放的是版权信息和联系方式等。

将上中下结构放置到一个DIV容器中，方便后面排版并且方便对页面进行整体调整。如图14-1所示。

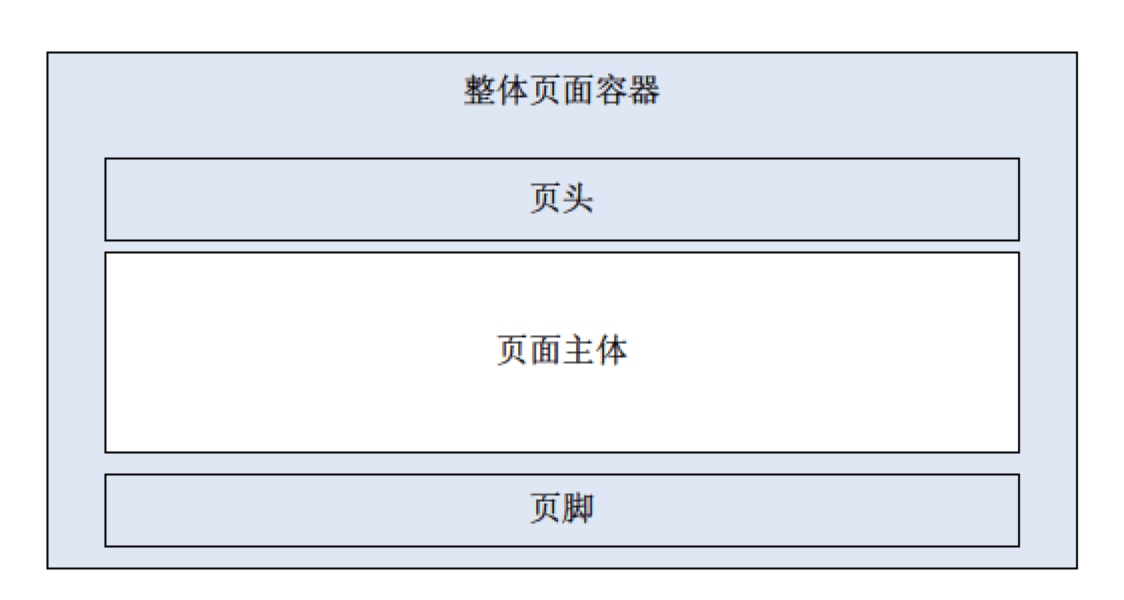


图14-1 上中下页面结构

14.1.2 设置各块位置

复杂的网页布局，不是单纯的一种结构，而是包含多种网页结构。例如，总体上是上中下，中间部分为两列布局等，如图14-2所示。

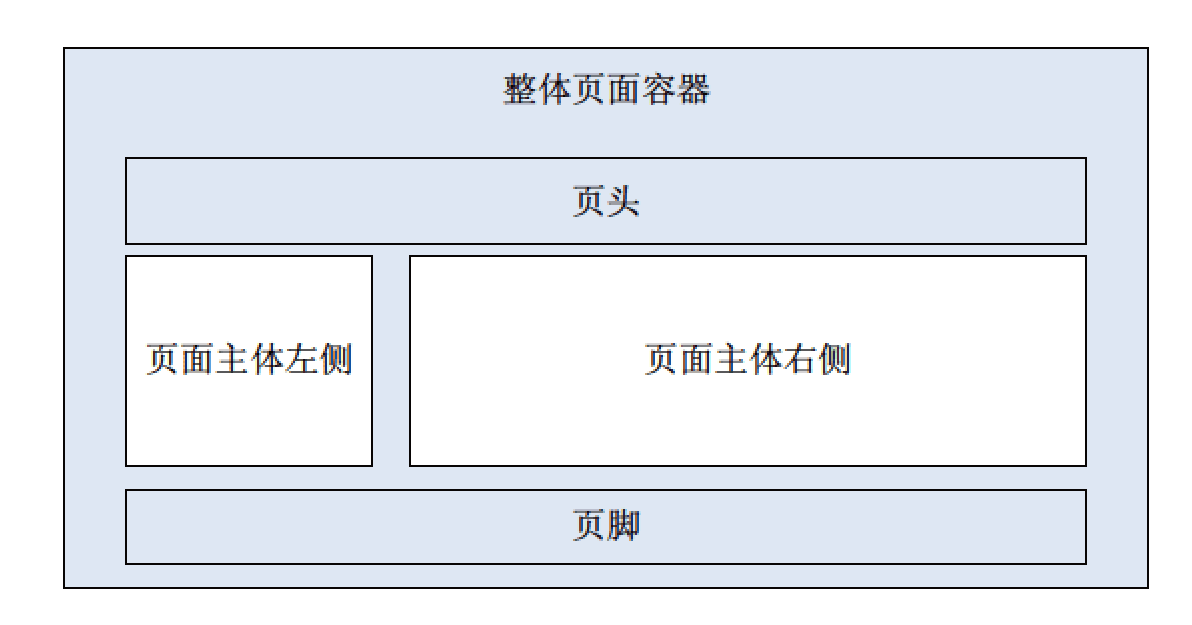


图14-2 复杂页面布局

页面总体结构确认后，一般情况下，页头和页脚变化就不大了。会发生变化的就是页面主体，此时需要根据页面展示的内容决定中间布局采用什么样式，三列水平分布还是两列分布等。

14.1.3 用CSS定位

页面版式确定后，就可以利用CSS对DIV进行定位，使其在指定位置出现，从而实现对页面的整体规划，然后再向各个页面添加内容。

本节将以图14-2为实现蓝图，创建一个总体为上中下布局、页面主体布局为左右布局的CSS定位实例。

【例14.1】实例文件：ch14\14.1.html

01 首先构建HTML页面，使用DIV划分最基本的布局块，其代码如下：

```
<html>
<head>
<title>CSS排版</title><body>
<div id="container">
  <div id="banner">页面头部</div>
  <div id=content >
  <div id="right">
页面主体右侧
  </div>
  <div id="left">
页面主体左侧
  </div>
</div>
  <div id="footer">页脚</div>
</div>
</body>
</html>
```

上面代码中，创建了5个层，其中ID名称为container的DIV层，是一个布局容器，即所有的页面结构和内容都是在这个容器内实现；名称为banner的DIV层，是页头部分；名称为footer的DIV层，是页脚部分。名称为content的DIV层，是中间主体，该层包含了两个层，一个是right层，一个是left层，分别放置不同的内容。

在IE9.0中浏览效果如图14-3所示，可以看到图中显示的这几层，从上到下依次排列。

02 对body标记和container层（布局容器）进行CSS修饰，从而对整体样式进行定义，代码如下：

```
<style type="text/css">
<!--
body {
  margin:0px;
  font-size:16px;
  font-family:"幼圆";
}
#container{
  position:relative;
  width:100%;
}
-->
</style>
```

上面代码只是设置了文字大小、字体、布局容器container的宽度、层定位方式，布局容器撑满整个浏览器。

在IE9.0中浏览效果如图14-4所示，可以看到此时相比较上一个显示页面，发生的变化不大，只不过字体和字体大小发生变化，因container没有边框和背景色而无法显示。

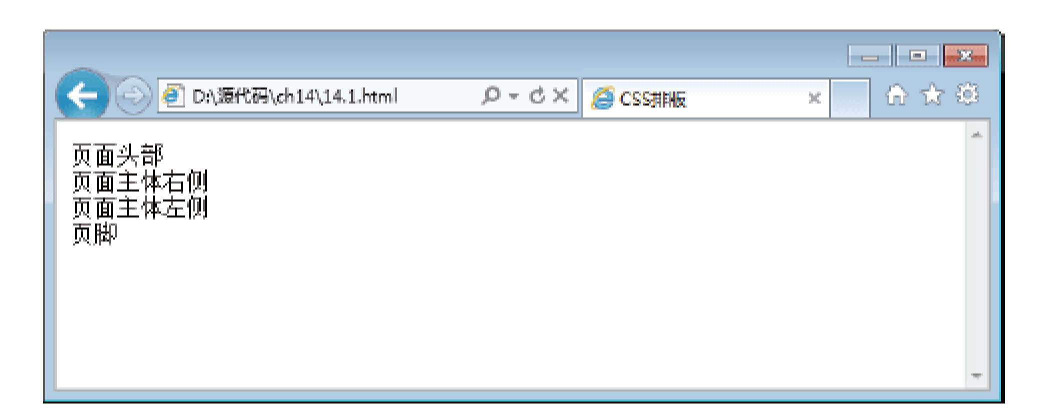


图14-3 创建DIV层

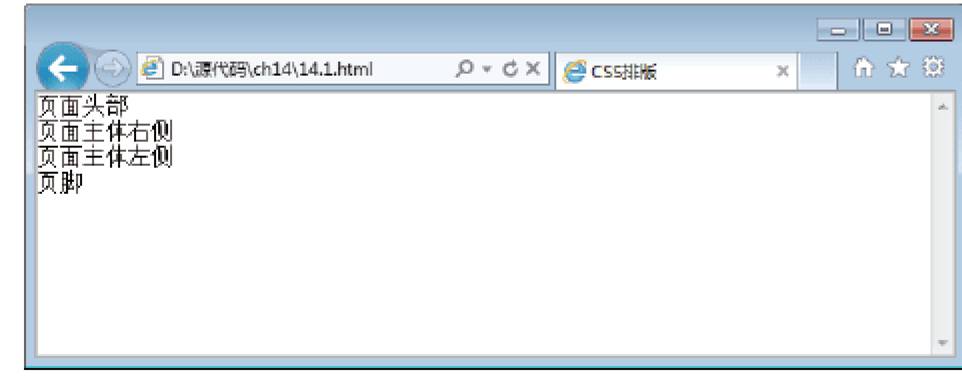


图14-4 设置全局CSS样式

03 使用CSS对页头进行定位，即banner层，使其在网页上显示，代码如下：

```
#banner{
  height:80px;
  border:1px solid #000000;
  text-align:center;
  background-color:#a2d9ff;
  padding:10px;
  margin-bottom:2px;
}
```

上面代码首先设置了banner层的高度为80px，宽度充满整个container布局容器，接着分别设置了边框样式、字体对齐方式、背景色、内边距和外边距的底部等。

在IE9.0中浏览效果如图14-5所示，可以看到在页面顶部显示了一

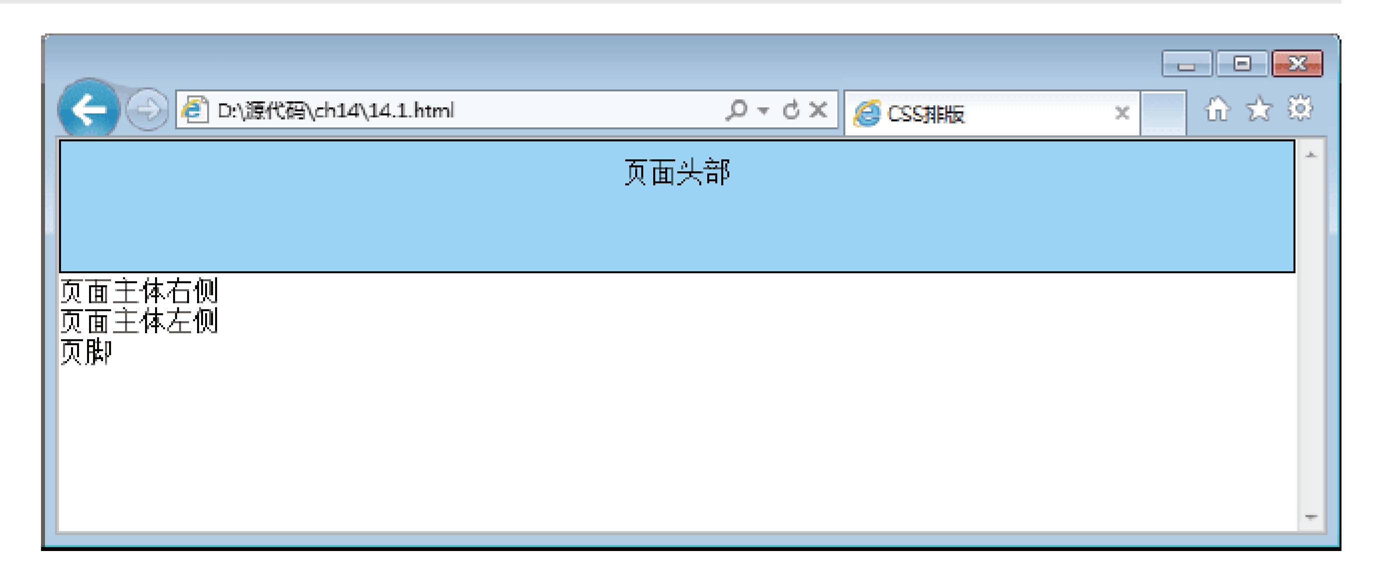


图14-5 CSS设置页头

个浅绿色的边框，边框充满整个浏览器，边框中间显示了一个“页面头部”的文本信息。

04 在页面主体如果两个层并列显示，需要使用float属性，将一个层设置到左边，一个层设置到右边。其代码如下：

```
#right{
  float:right;
  text-align:center;
  width:80%;
 border:1px solid #ddeecc;
margin-left:1px;
height:200px;
}
#left{
  float:left;
  width:19%;
  border:1px solid #000000;
  text-align:center;
height:200px;
background-color:#bcbcbc;
}
```

上面代码设置了这两个层的宽带，right层占有空间的80%，left层占有空间的19%，并分别设置了两个层的边框样式、对齐方式、背景色等。

在IE9.0中浏览效果如图14-6所示，可以看到页面主体部分，分为两个层并列显示，左边背景色为灰色，占有空间较小，右侧背景色为白色，占有空间较大。

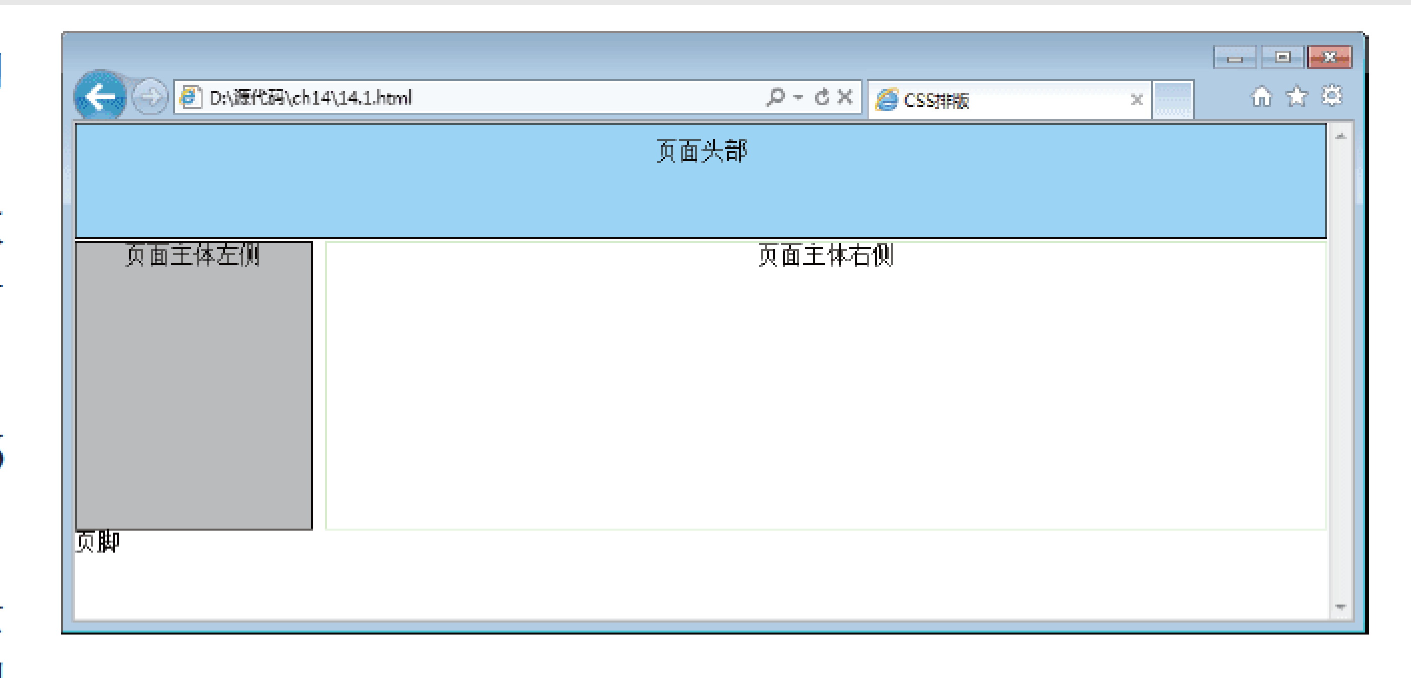


图14-6　CSS定位中间主体

05 最后需要设置页脚部分，页脚通常在主体下面。因为页面主体中使用了float属性设置层浮动，所以需要在页脚层设置clear属性，使其不受浮动的影响。其代码如下：

```
#footer{
  clear:both;         /* 不受float影响 */
  text-align:center;
  height:30px;
  border:1px solid #000000;
               background-color:#ddeecc;
}
```

上面代码设置页脚对齐方式、高度、边框和背景色等。在IE9.0中浏览效果如图14-7所示，可以看到页面底部显示了一个边框，背景色为浅绿色，边框充满整个DIV布局容器。

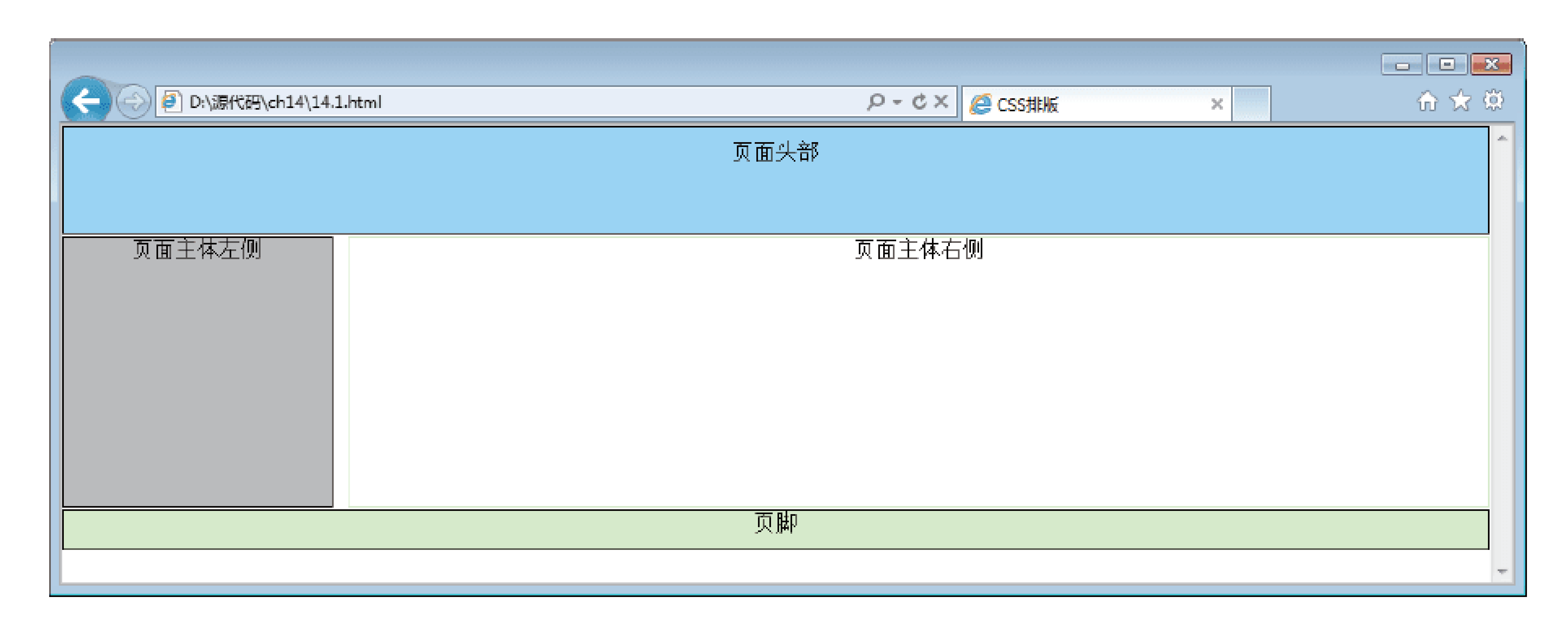


图14-7 CSS定义页脚



14.2 固定宽度布局

在进行网页设计时，必须考虑网页布局宽度，因为浏览者会使用大小不同的浏览器查看页面，设计宽度过大，有的浏览器就不能完全显示出来；设计宽度过小，显示内容就不能完成显示并容易出现滚动条。

14.2.1 上中下版式

网页开发过程中，有几种比较经典的网页排版方式，包括宽度固定的上中下版式、宽度固定的左右版式、自适应宽度布局和浮动布局等。这些版式会经常网页设计时出现，并且经常用在各种类型的网站开发中。

宽度固定的上中下版式布局，应用得非常广泛。但其一般用在简单信息介绍上，而不用在网站首页上。这种版式布局简单，包含信息较少。本节将介绍一个上中下版式布局，其效果如图14-8所示。

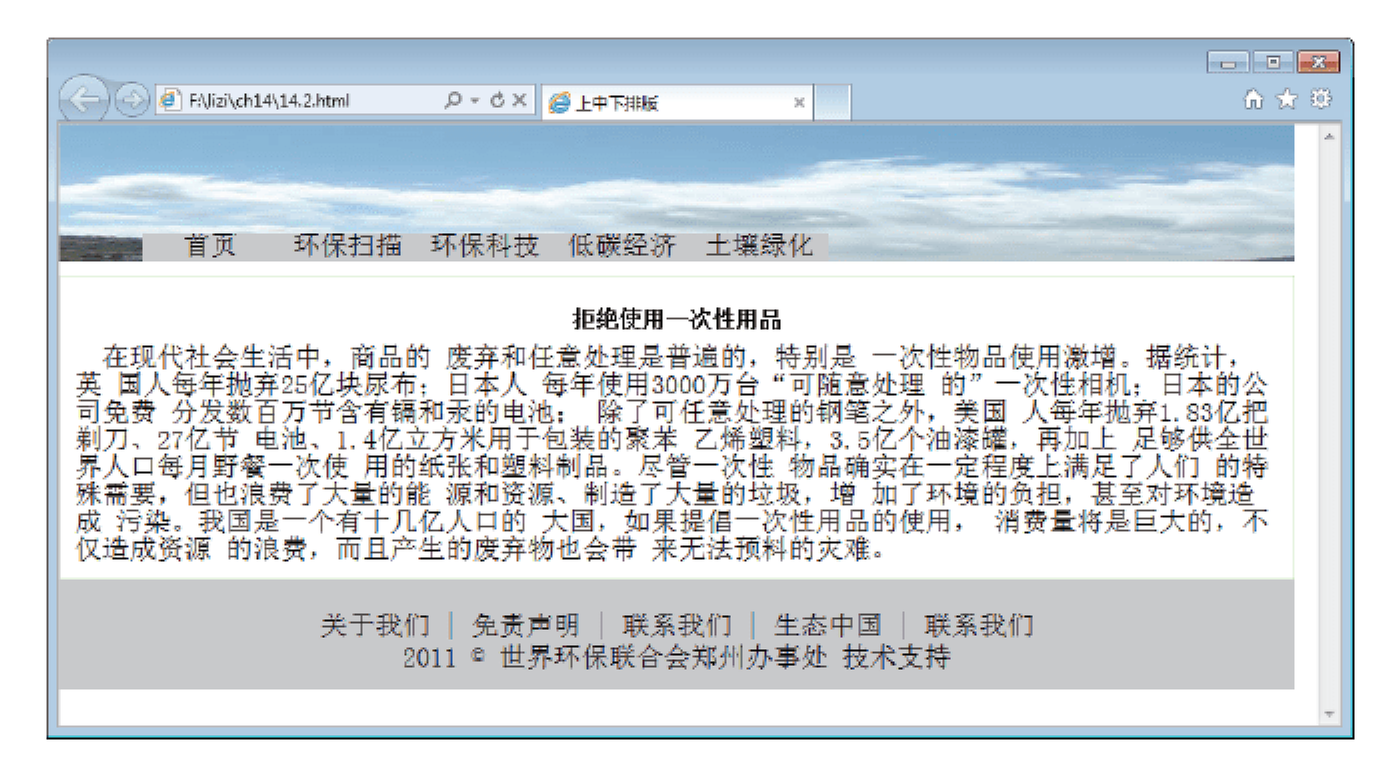


图14-8 上中下布局

从上面的效果可以看到，这个页面一共分为三个部分，第一部分包含图片和菜单栏，这一部分放到页头，是上中下版式的“上”；第二个部分是中间的内容部分，即页面主体，用于存放要显示的文本信息，是上中下版式的“中”；第三个部分是页面底部，包含地址和版权信息的页脚，是上中下版式的“下”。

需要说明的是，本节主要介绍网页布局，所以对其他样式修饰的CSS代码，这里就不再介绍了，读者可以查看光盘中提供的相关代码。

【例14.2】实例文件：ch14\14.2.html

01 使用DIV块对页面区域进行划分，使其符合如图14-8所示的页面布局，基本代码如下：

```
    <html>
    <head>
    <title>上中下排版</title>
    </head>
    <body>
      <div class="big">
          <div class="up">
              <p><a href="#">首页</a><a href="#">环保扫描</a><a href="#">环保
科技</a><a href="#">低碳经济</a><a href="#">土壤绿化</a></p></div> <div class="middle">
              <br />
              <h1>拒绝使用一次性用品</h1>
           <p>  在现代社会生活中，商品的 废弃和任意处理是普遍的，特别是 一次性物品使用
激增。据统计，英国人每年抛弃25亿块尿布……
    </p>
            </div>
            <div class="down">
            <br />
              <p><a href="#">关于我们</a> | <a href="#">免责声明</a> | <a
href="#">联系我们</a> | <a href="#">生态中国</a> | <a href="#">联系我们</a></p>
                <p>2011 &copy; 世界环保联合会郑州办事处 技术支持</p>
            </div>
        </div>
    </body>
    </html>
```

上面代码创建了4个层，big层是DIV布局容器，用来存放其他DIV块；up层表示页头部分；middle层表示页面主体；down层表示页脚部分。

在IE9.0中浏览效果如图14-9所示，可以看到页面显示了3个区域信息，顶部显示的是超级链接部分，中间显示的是段落信息，底部显示的是地址和版权信息。其布局从上到下自动排列，不是期望的布局方式。

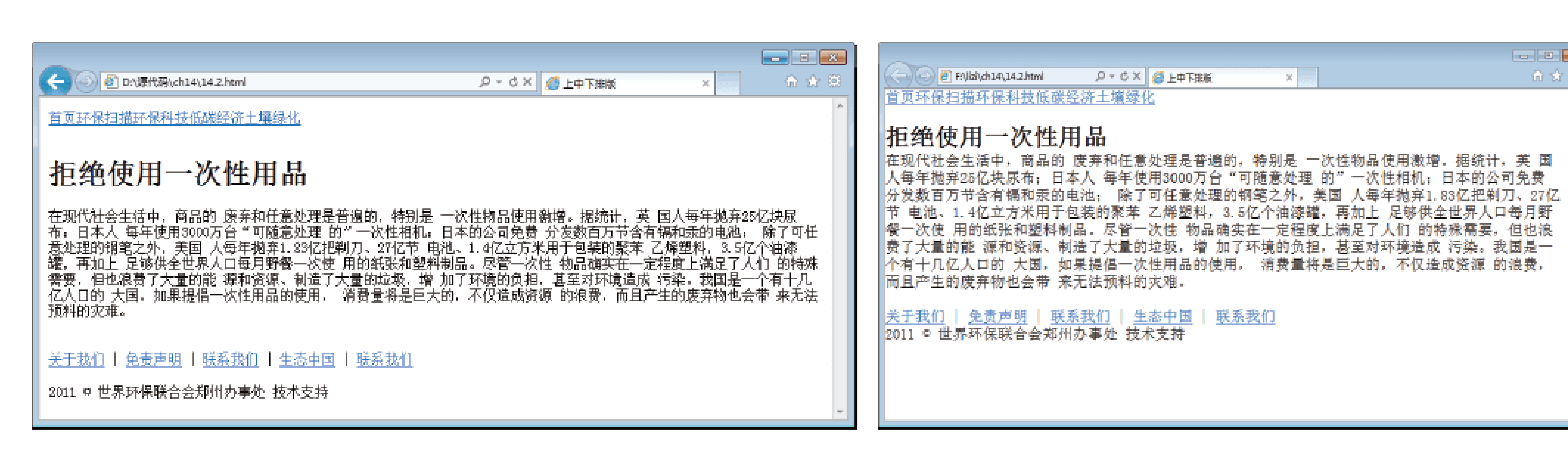


图14-9 实现DIV块　　　　图14-10 定义整体样式

02 上面页面显示时，字体样式非常丑陋，布局不合理。此时需要使用CSS代码，对页面整体样式，进行修饰。其代码如下：

```
<style>
  *{
    padding:0px;
    margin:0px;
    }
  body{
    font-family:"幼圆";
    font-size:12px;
     color:green;
    }
  .big{
    width:900px;
    margin:0 auto 0 auto;
    }
</style>
```

上面代码定义了页面整体样式，字体为“幼圆”，字体大小为12px，字体颜色为绿色，布局容器big的宽度为900px。“margin:0 auto 0 auto”语句表示该块与页面的上下边界之间的距离为0，左右自动调整。

在IE9.0中浏览效果如图14-10所示，可以看到页面字体变小，字体颜色为绿色，并充满整个页面，页面宽度为900px。

03 使用CSS定义页头部分，即导航菜单，其代码如下：

```
  .up{
    width:900px;
    height:100px;
    background-image:url(17.jpg);
    background-repeat:no-repeat;
    }
```

在类选择器up中，CSS定义层的宽度和高度，其宽度为900px，并定义了背景图片。

在IE9.0中浏览效果如图14-11所示，可以看到页面顶部显示了一个背景图，并且超级链接以一定距离显示，以绝对定位方式在页头显示。

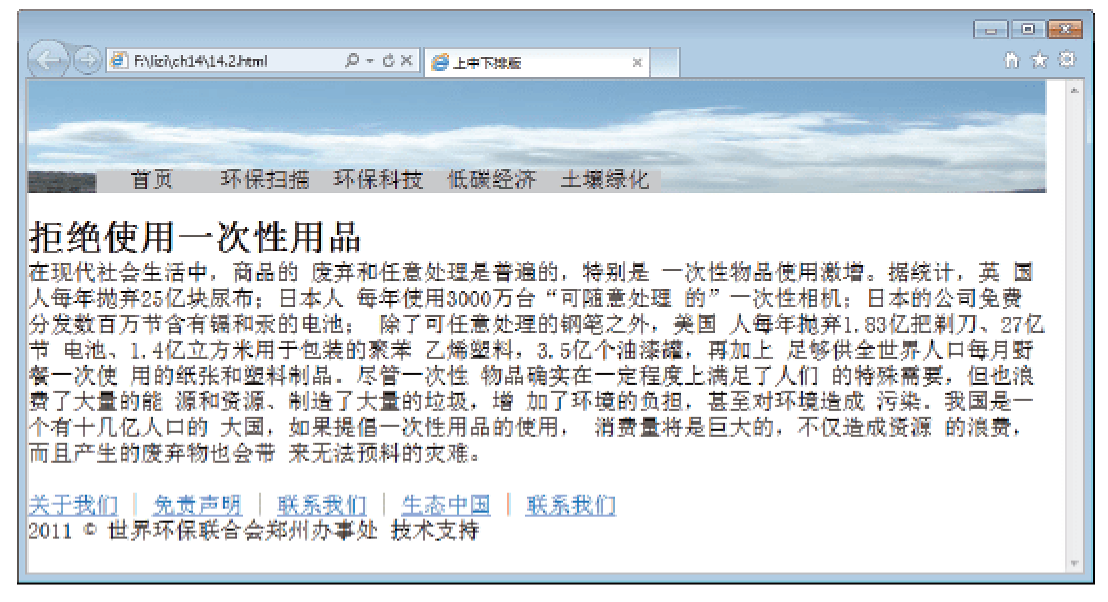


图14-11 CSS修饰页头

图14-12 CSS定义中间部分

04 使用CSS定义页面主体，即定义层和段落信息，代码如下：

```
  .middle{
    border:1px #ddeecc solid;
    margin-top:10px;
```

```
      }
```

在类选择器middle中，定义了边框样式和内边距，此处层的宽度和big层宽度一致。

在IE9.0中浏览效果如图14-12所示，可以看到中间部分以边框形式显示，标题居中显示，段落缩进两个字符显示。

05 定义页脚部分代码如下所示：

```
.down{
  background-color:#CCCCCC;
  height:80px;
  text-align:center;
  }
```

上面代码中，类选择器down定义了背景颜色、高度和对齐方式。其他选择器定义超级链接的样式。

在IE9.0中浏览效果如图14-8所示，可以看到页面底部显示了一个灰色矩形框，其版权信息和地址信息居中显示。

14.2.2 左右版式

在页面排版中，有时根据内容需要页面主体会分为左右两个部分显示，用来存放不同的信息内容。实际上这也是一种宽度固定的版式。

需要说明的是，本节主要介绍网页布局，所以对其他修饰的CSS样式代码，这里就不再介绍了，读者可以查看光盘相关内容。

本节实例完成后的效果如图14-13所示。此框架大体上也是上中下结构，只不过页面主体部分又分了左右版式。



图14-13 左右版式效果

【例14.3】实例文件：ch14\14.3.html

01 在HTML页面，将DIV框架和所要显示的内容显示出来，并将要引用的样式名称定义好，代码如下：

```
<html>
<head>
<title>茶网</title>
   </head>
```

```
<body>
<div id="container">
  <div id="banner">
    <img src="b.jpg" border="0">
  </div>
  <div id="links">
    <ul>
      <li>首页</li>
      <li>茶业动态</li>
      <li>名茶荟萃</li>
      <li>茶与文化</li>
      <li>茶艺茶道</li>
      <li>鉴茶品茶</li>
      <li>茶与健康</li>
     <li>茶语清心</li>
    </ul>
    <br>
  </div>
  <div id="leftbar">
      <p class="lefttitle">名人与茶</p>
      <p>.三文鱼茶泡饭</p>
       <p>.董小宛的茶泡饭</p>
       <p>.人生百味一盏茶</p>
      <p>.我家的茶事</p>
    <p class="lefttitle">茶事掌故</p>
     <p>.“峨眉雪芽”的由来</p>
    <p>.茶文化的养生术</p>
     <p>.老北京的花茶</p>
    <p>.古代洗茶的原因和来历</p>
  </div>
  <div id="content">
    <h4>人生茶境</h4>
    <p>
“喝茶当于瓦纸窗下，清泉绿茶，用素雅的陶瓷茶具，同二三人共饮，得半日之闲，可抵十年的尘梦。”
</p>
<p>
对中国人来说，“茶”是一个温暖的字……
</p>
  </div>
  <div id="footer">版权所有 2011.08.12</div>
</div>
</body>
</html>
```

上面代码定义了几个层用来构建页面布局。其中层container作为布局容器，banner作为页面图形Logo，links层作为页面导航，leftbar层作为左侧内容部分，content层作为右侧内容部分，footer层作为页脚部分。

在IE9.0中浏览效果如图14-14所示，可以看到页面上部显示了一张图片，下面是超级链接和段落信息，最后是地址信息等。

02 定义整体样式，例如网页中字形或对齐方式等，代码如下：

```
<style>
<!--
body, html{
  margin:0px; padding:0px;
  text-align:center;
}
#container{
  position: relative;
  margin: 0 auto;
  padding:0px;
  width:700px;
  text-align: left;
}
-->
</style>
```

上面代码中，类选择器container定义了布局容器的定位方式为相对定位，宽度为700px，文本左对齐，内外边距都为0px。

在IE9.0中浏览效果如图14-15所示，可以看到与上一个页面比较，发生变化不大。

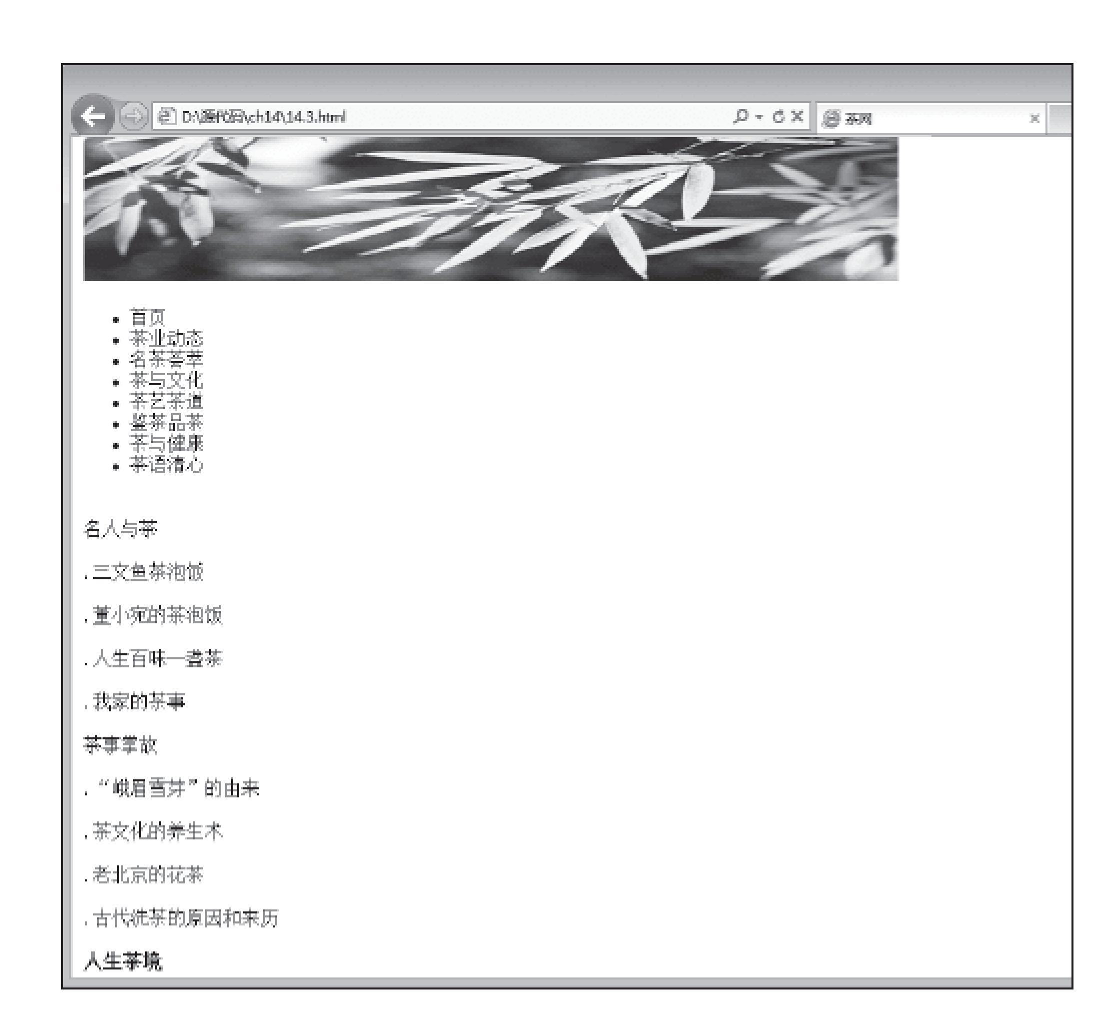


图14-14 DIV层构建布局样式

图14-15 整体样式定义

03 此网页的页头部分，包含两个部分，即Logo和导航菜单。这两个层的CSS代码如下：

```
#banner{
  margin:0px; padding:0px;
}
#links{
  font-size:12px;
  margin:-18px 0px 0px 0px;
  padding:0px;
  position:relative;
}
```

上面代码中，ID选择器banner定义了内外边距都是0px，ID选择器links定义了导航菜单的样式，例如字体大小为12px，定位方式为相对定位等。

在IE9.0中浏览效果如图14-16所示，可以看到页面导航部分在图像上显示，并且每个菜单相隔一定距离。

04 使用CSS代码定义页面主体左侧部分，代码如下：

```
#leftbar{
  background-color:#d2e7ff;
  text-align:center;
  font-size:12px;
  width:150px;
float:left;
  padding-top:0px;
  padding-bottom:30px;
  margin:0px;
}
```

上面选择器leftbar中定义了层背景色、对齐方式、文字大小和左侧DIV层的宽度，这里使用float定义层在水平方向上浮动定位。

在IE9.0中浏览效果如图14-17所示，可以看到页面左侧部分以矩形框显示，包含了一些简单的页面导航。

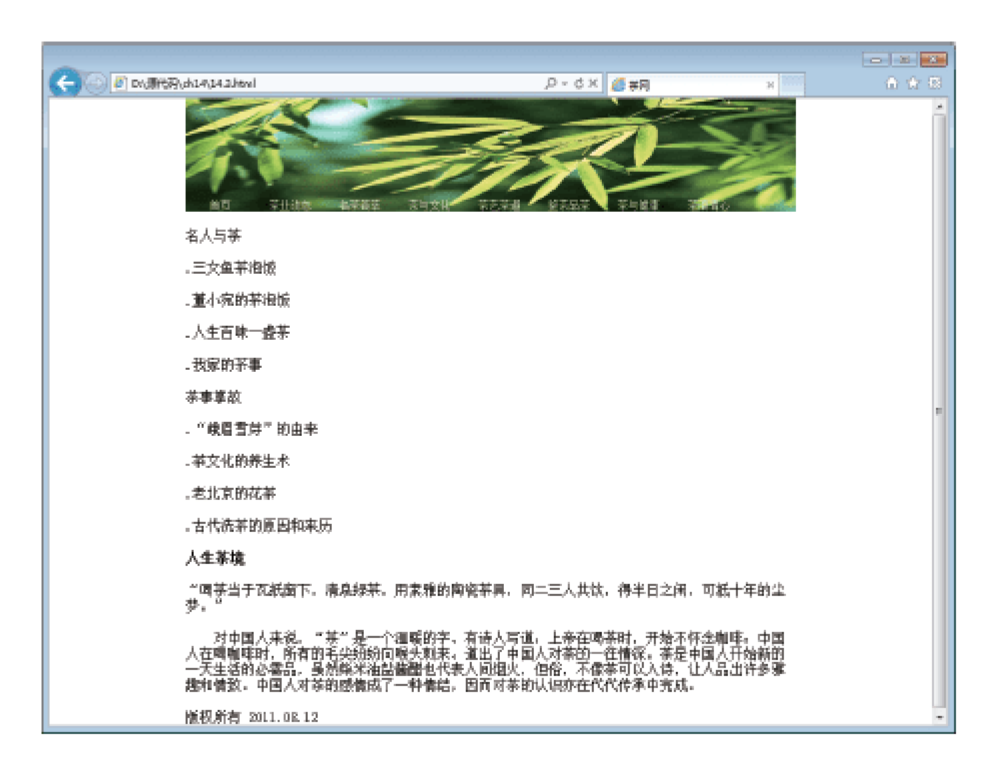


图14-16 定义页头部分



图14-17 CSS定义页面左侧

05 使用CSS代码定义页面主体右侧部分，代码如下：

```
#content{
  font-size:12px;
  float:left;
width:550px;
  padding:5px 0px 30px 0px;
  margin:0px;
}
```

代码中ID选择器content用来定义字体大小、右侧div层宽度，内外边距等。在IE9.0中浏览效果如图14-18所示，可以看到右侧部分的段落字体变小，段落缩进了两个单元格。

06 如果上面的层使用了浮动定位，页脚一般需要使用clear去掉浮动所带来的影响，其代码如下所示：

```
#footer{
  clear:both;
font-size:12px;
  width:100%;
  padding:3px 0px 3px 0px;
  text-align:center;
  margin:0px;
  background-color:#b0cfff;
}
```

footer选择器中定义了层的宽度、字体、对齐方式和背景色。在IE9.0中浏览效果如图14-13所示，可以看到页脚显示了一个矩形框，背景色为浅蓝色，矩形框内显示了版权信息。



图14-18 定义右侧内容



14.3 自适应宽度布局

一般的门户网站都采用了一种典型的960px的固定宽度布局，但现在的浏览器分辨率越来越大，有时候需要尽量利用屏幕的空间，可以采用自适应布局模式。

本节创建了一个中间宽度自适应，左右两侧宽度固定的页面，如图14-19所示。

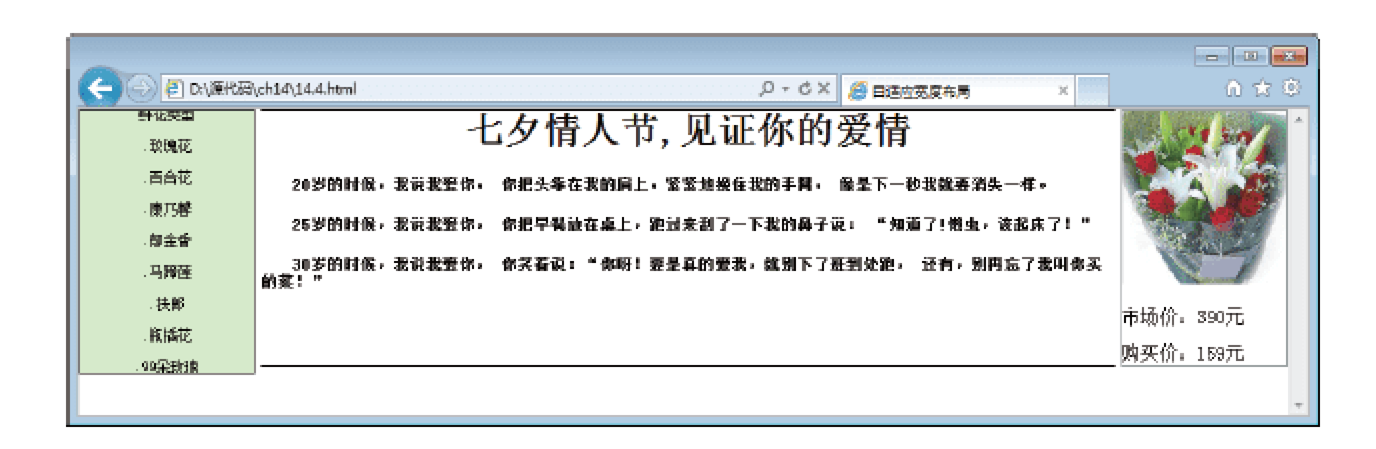


图14-19 自适应宽度布局

上面页面可以划分为3个部分，分别是左、中、右布局。左侧存放列表，中间存放段落，右侧存放图片信息等。

【例14.4】实例文件：ch14\14.4.html

01 构建HTML，使用DIV层布局。

在HTML页面中使用DIV层，将页面划分为不同的区域，即划分左侧层、右侧层和中间层，其代码如下：

```
<html>
<head>
<title>自适应宽度布局</title>
</head>
<body>
<div id="wrap">
<div class="wrap_l">
<p>鲜花类型</p>
<p>.玫瑰花</p>
<p>.百合花</p>
<p>.康乃馨</p>
<p>.郁金香</p>
<p>.马蹄莲</p>
<p>.扶郎</p>
<p>.瓶插花</p>
<p>.99朵玫瑰</p>
</div>
<div class="wrap_r">
<img src="41.jpg"/>
<p>市场价：390元</p>
<p>购买价：159元</p>
</div>
<div class="wrap_m">
<h1>七夕情人节,见证你的爱情</h1><p>
20岁的时候，我说我爱你，你把头靠在我的肩上，紧紧地挽住我的手臂，像是下一秒我就要消失一样。
</p></div></div>
</body>
</html>
```

上面代码定义了4个层，层wrap是个布局容器，用来存放页面内所有的文本和图片信息。wrap_l层定义页面左侧部分，wrap_r定义页面右侧部分，wrap_m定义页面中间部分。

在IE9.0中浏览效果如图14-20所示，可以看到页面自上而下显示列表、图片和文本信息等。

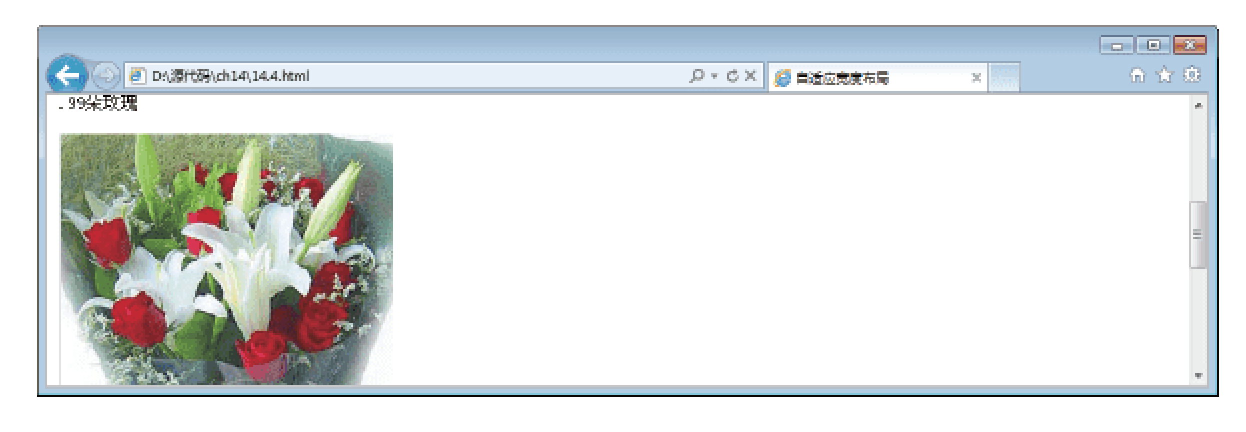


图14-20 div层构建布局

02 CSS设置页面整体样式。

首先设置页面整体样式，如字体、背景色等。其代码如下：

```
<style type="text/css">
<!--
body{margin:0;padding:0px;text-align:center;}
#wrap{margin:0 auto;text-align:left;}
</style>
```

上面代码定义了body页面和层wrap的内外边距和对齐方式。页面没有发生较大的变化，这里就不再显示了。

03 CSS定义左侧内容。

```
/*左边栏，固定宽度*/
.wrap_l{float:left;margin-right:-150px;width:150px;
border:1px solid #333;background-color:#ddeecc;margin-
right:2px;height:220px;}
.wrap_l p{
      text-align:center;
       font-family:"幼圆";
       font-size:13px;
       line-height:8px;
}
```

上面代码类选择器wrap_l定义了页面左侧部分的宽度为150px，高度为220px，右侧内边距为-150px，页面浮动在左侧显示以及边框样式、背景色等。下面的选择器定义段落的显示样式，例如字体大小、字体、对齐方式和行高等。

在IE9.0中浏览效果如图14-21所示，可以看到页面左侧显示了一个矩形框，其背景色为浅蓝色并居中对齐。

04 CSS定义中间内容。

刚才左侧部分定义了固定的宽度，而中间部分就不需要定义宽度了，其CSS代码如下：

```
/*中间栏，自适应宽度*/
.wrap_m{width:auto;margin:0 140px 0 150px;border-top:1px solid #000;
border-bottom:1px solid #000;height:220px;}
.wrap_m h1{
      text-align:center;
}
.wrap_m p{
        font-size:13px;
        font-weight:bolder;
        text-indent:2em;
        font-family:"幼圆";
}
```

上面代码中，wrap_m类选择器定义中间部分显示样式，例如宽度自适应、外边距大小、层的上边框样式等。下面的选择器定义了字体样式。

在IE9.0中浏览效果如图14-22所示，可以看到页面中间部分还是在页面底部显示，其字体加黑变粗显示，段落缩进两个字符显示，标题居中显示。

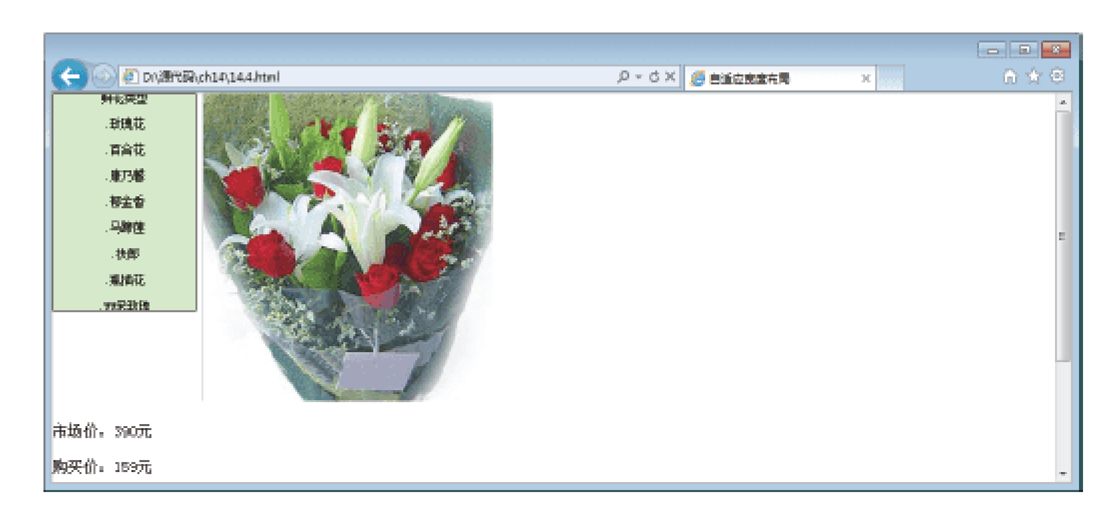


图14-21 定义页面左侧部分

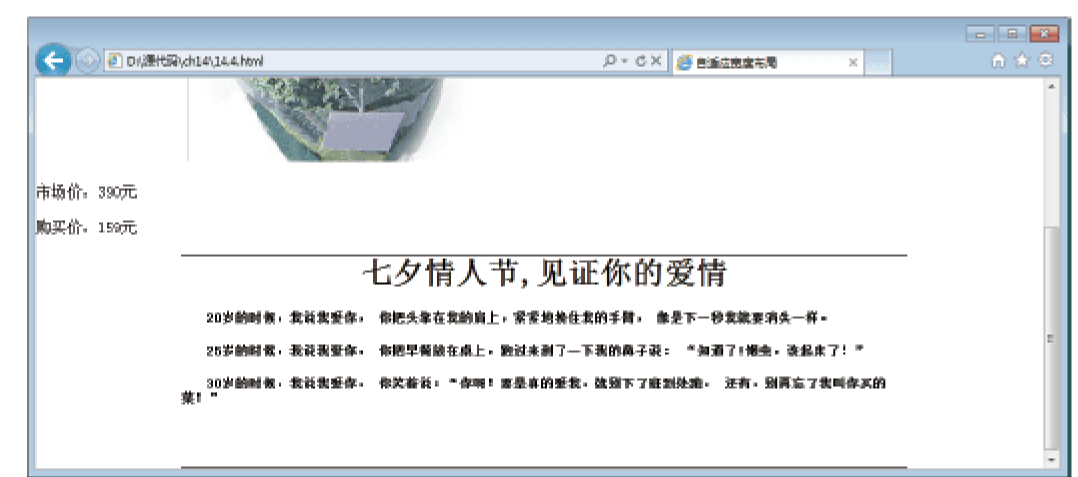


图14-22 定义中间部分

05 CSS定义右侧内容。

右侧部分需要固定页面宽度，其CSS代码如下：

```
    /*右边栏，固定宽度*/
    .wrap_r{float:right;margin-left:-140px;width:140px;border:1px solid
#999;margin-left:2px;height:220px;}
    img{width:140px;}
    .wrap_r p{
       line-height:12px;
       font-family:"幼圆";
    }
```

在wrap_r类选择器中，float属性定义了层浮动在右边显示，宽度为140px，高度为220px等。img选择器中定义了图片宽度，最后一个选择器定义了行高和字形。

在IE9.0中浏览效果如图14-19所示，可以看到页面3个层在页面并列显示，其中右侧部分显示了一个图片。读者可以自由调整页面宽度，中间宽度会随着浏览器的大小自动发生变化。



14.4 其他页面排版

除了上面固定宽度布局和自适应布局外，还可以利用前面介绍的float和display属性，分别实现常见的浮动布局和隐藏布局。

14.4.1 浮动布局

DIV+CSS网页布局常用到浮动布局，但浮动并不像表格那样好用，很多时候会出问题。同时设计不够良好的浮动布局，在不同的浏览器下会有不同的表现。对浮动布局支持比较好的是Firefox浏览器。

本节创建一个浮动布局，其效果如图14-23所示。从页面效果上可以看出当前页面分为两个层，一个是左侧层，一个是右侧层。

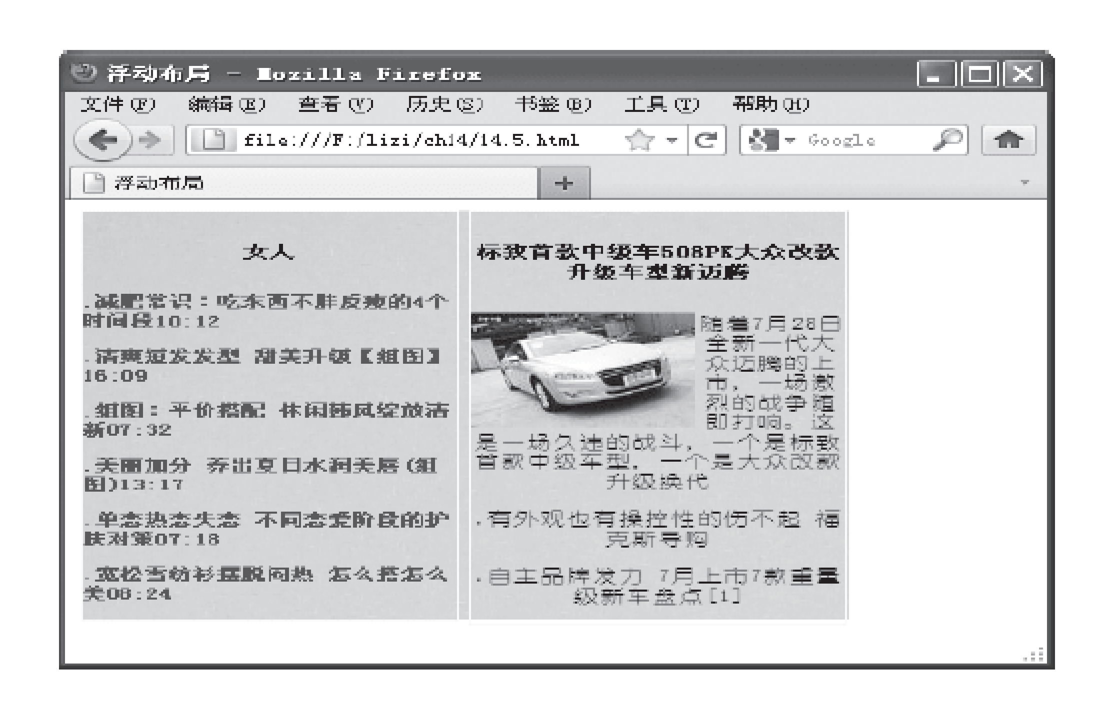


图14-23 浮动布局

【例14.5】实例文件：ch14\14.5.html

01 创建HTML页面，使用DIV层布局。

在HTML页面中，使用DIV将页面划分为左侧和右侧两个部分，并将这两个部分放入到一个DIV层，方便页面布局和排版，代码如下：

```
<html>
<head>
<title>浮动布局</title>
</head>
<body>
<div id="wrap" >
<div id="col1" >
<h5>女人</h5>
<p>.减肥常识：吃东西不胖反瘦的4个时间段10:12</p>
<p>.清爽短发发型  甜美升级【组图】16:09 </p>
<p>.组图：平价搭配  休闲韩风绽放清新07:32 </p>
<p>.美丽加分  养出夏日水润美唇(组图)13:17</p>
<p>.单恋热恋失恋  不同恋爱阶段的护肤对策07:18</p>
<p>.宽松雪纺衫摆脱闷热  怎么搭怎么美08:24 </p>
</div>
<div id="col2" >
<h5>标致首款中级车508PK大众改款升级车型新迈腾</h5>
<img src=1.jpg/>
<p>随着7月28日全新一代大众迈腾的上市，一场激烈的战争随即打响。这是一场久违的战斗，一个是标致首款中级车型，一个是大众改款升级换代</p>
<p>.有外观也有操控性的伤不起  福克斯导购 </p>
<p>.自主品牌发力  7月上市7款重量级新车盘点[1] </p>
</h3>
</div>
</div>
</div>
</body>
```

上面代码中，创建了3个层，层wrap作为布局容器，col1作为左侧布局层，col2作为右侧布局层。

在IE9.0中浏览效果如图14-24所示，可以看到页面自上而下显示信息，页面布局非常混乱。

02 CSS定义整体样式。

使用CSS代码定义页面整体样式，即wrap中的所有元素，代码如下：

```
<style>
#wrap{
height:300px;
width:410px;
background:url(87.JPG) repeat-y left top;
}
</style>
```

上面代码定义了wrap层，宽度为410px，高度为300px，并设置背景图片，图片在y轴重复。

在Firefox 5.0中浏览效果如图14-25所示，可以看到页面显示一个背景图片为蓝色的区域，该区域用来存放左右两侧内容。

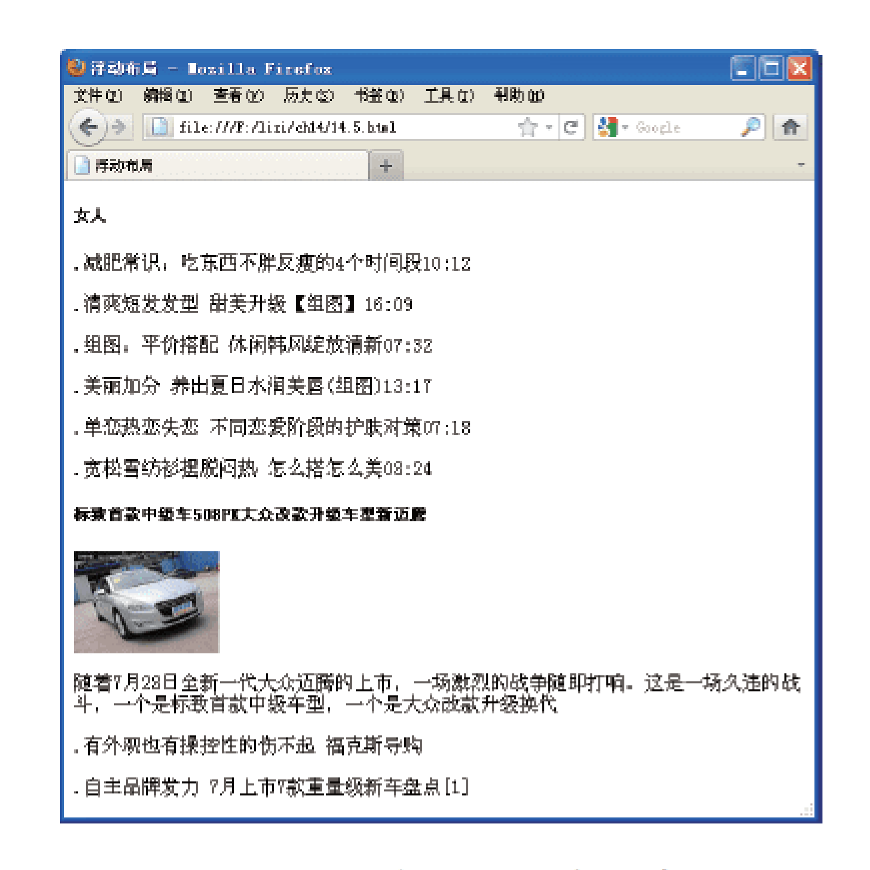


图14-24 使用DIV构建布局

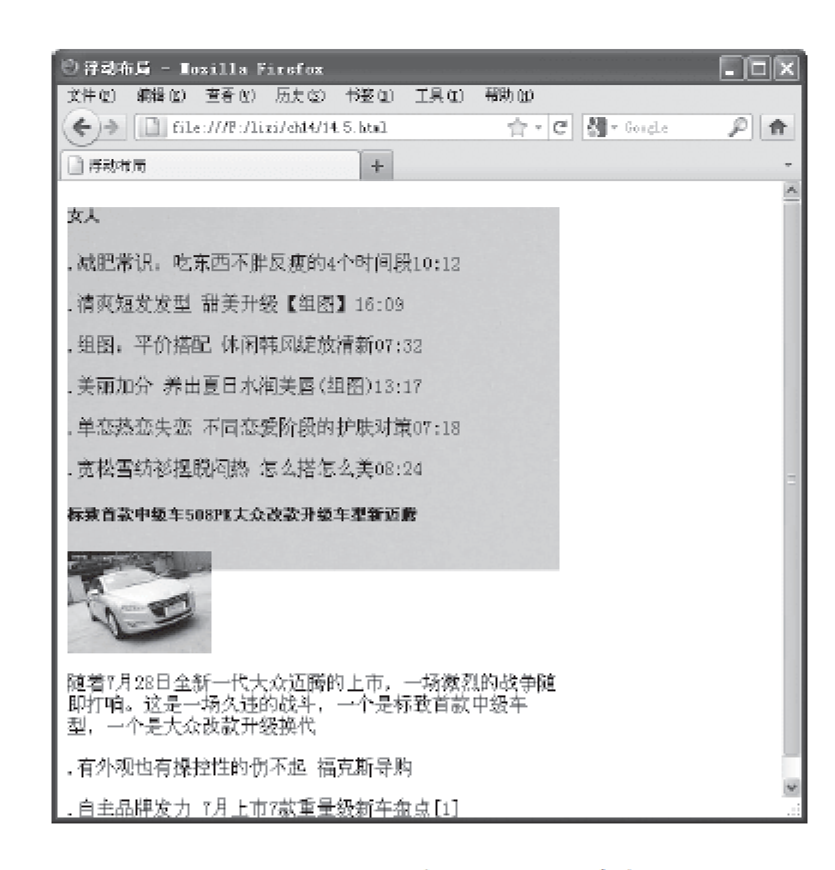


图14-25 定义wrap样式

03 CSS定义左侧部分。

使用CSS定义页面左侧部分，代码如下：

```
#col1{
float:left;
width:200px;
text-align:center;
border:1px solid #ddeecc;
margin-right:5px;
}
#col1 p{
   text-align:left;
    font-size:12px;
    color:#123456;
    font-weight:bolder;
}
```

上面代码中，ID选择器col1定义了层在左侧悬浮显示，宽度为200px，居中对齐，边框以直线型显示等。下面的选择器定义了字体样式，例如对齐方式、字体大小、字体颜色和字体样式等。

在Firefox 5.0中浏览效果如图14-26所示，可以看到页面蓝色区域中，左侧部分以列表形式显示页面导航链接。

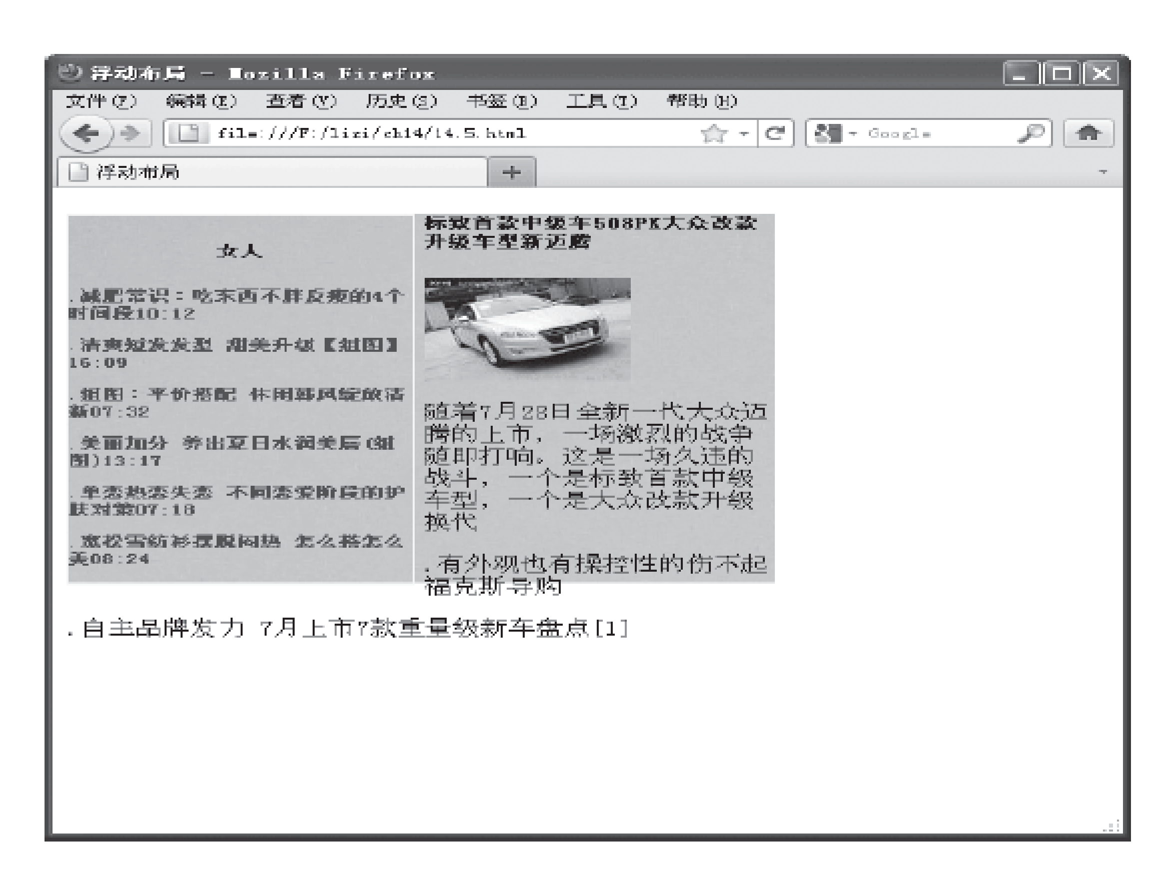


图14-26 定义左侧样式

04 CSS定义右侧部分。

使用CSS代码，定义页面右侧部分样式，代码如下：

```
#col2{
float:left;
width:200px;
text-align:center;
border:1px solid #ddeecc;
}
img{
    float:left;
}
#col2 p{
      font-size:13px;
      font-family:"幼圆";
      line-height:14.5px;
}
```

ID选择器col2中，定义了层的悬浮方式、宽度、对齐方式和边框样式，img选择器中定义图片悬浮方式，最后一个选择器定义了字体大小、字体和行高等。

在Firefox 5.0中浏览效果如图14-23所示，可以看到左右两侧都在蓝色区域内显示，右侧部分显示了一个图片，其文字环绕图片显示。

14.4.2 隐藏布局

在有的网站，有时根据需要会自动或手动隐藏一些层，从而为其他层节省显示空间，这种方式称为隐藏布局。实现层的手动隐藏或展开，需要CSS代码和JavaScript代码相结合来实现。

因为涉及了JavaScript代码，本节创建了一个简单隐藏布局，用以展示隐藏布局的特点。其效果如图14-27所示。

【例14.6】实例文件：ch14\14.6.html

01 构建HTML代码。

在HTML网页中，创建了一个DIV层，用来包含要显示的信息，代码如下：

```
<html>
<head>
<title>隐藏布局</title>
</head>
<body>
<a href="#" onclick="toggle('div1')">显示/隐藏</a>
<div id="div1" class="div">
<img src=49.jpg>
<p>市场价：390元</p>
<p>购买价：190元</p>
</div>
</body>
</html>
```

在代码中，创建了一个超级链接和一个DIV层div1，DIV层中包含了图片和段落信息。在IE9.0中浏览效果如图14-28所示，可以看到页面显示了一个超级链接，超级链接下方显示了一张图片，图片下方是段落信息。

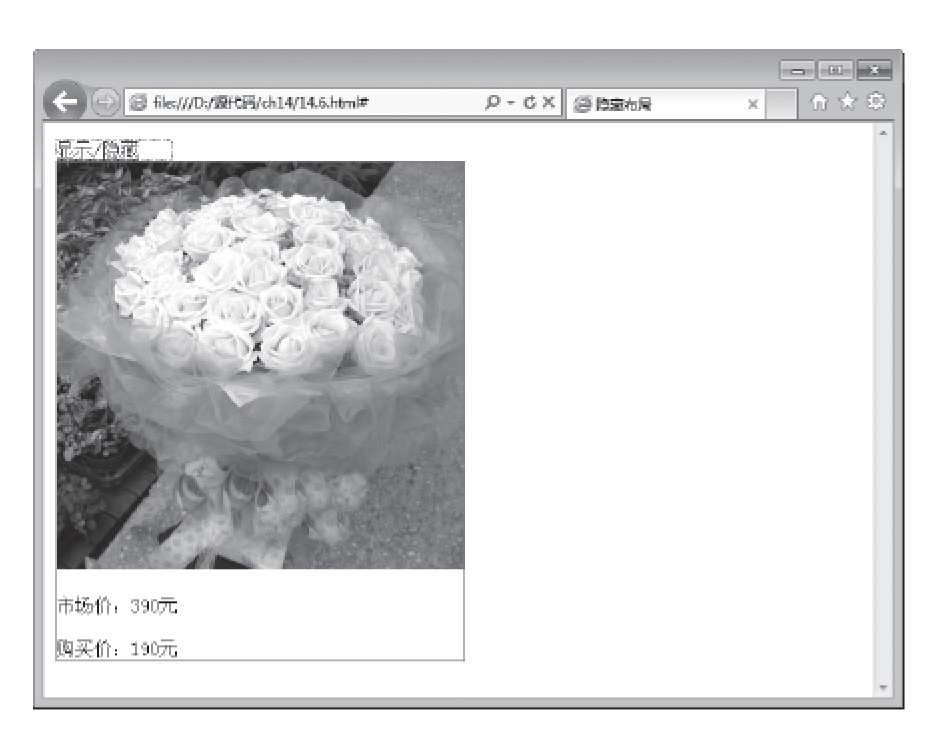


图14-27 隐藏布局



图14-28 设置div层

02 添加CSS代码。

使用CSS设置DIV层的显示方式，代码如下：

```
<style type="text/css">
.div{ border:1px #06F solid;height:50px;width:150px;display:none;}
a {width:100px; display:block}
</style>
```

在类选择器div中，定义了边框样式、高度和宽度，并使用display属性设定层不显示。在IE9.0中浏览效果如图14-29所示，可以看到只显示了一个超级链接，而没有其他信息。

03 添加JavaScript实现层隐藏。

下面需要添加JavaScript代码，实现层的手动显示和隐藏，代码如下：

```
<script language="JavaScript" type="text/JavaScript">
```

```
<!--
function toggle(targetid){
     if (document.getElementById){
         target=document.getElementById(targetid);
             if (target.style.display=="block"){
                 target.style.display="none";
             } else {
                 target.style.display="block";
             }
     }
}
-->
</script>
```

上面代码首先根据ID名称targetid，判断display的当前属性值，如果值为block，则设置为none；如果值为none，则设置值为block。

在IE9.0中浏览效果如图14-27所示，单击【显示/隐藏】超级链接，会自动显示图片相关信息。

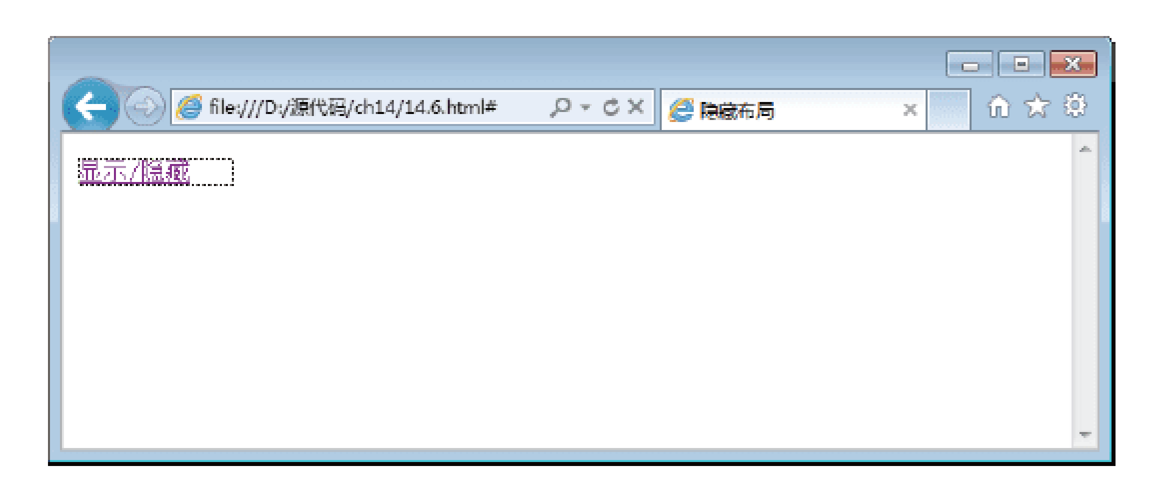


图14-29 设定CSS样式



14.5 新增CSS3多列布局

在CSS3没有出来之前，网页设计者如果要设计多列布局，不外乎有两种方式，一种是浮动布局，另一种是定位布局。浮动布局比较灵活，但容易发生错位，需要添加大量的附加代码或无用的换行标记，增加了不必要的工作量。定位布局可以精确地确定位置，不会发生错位，但无法满足模块的适应能力。为了解决多列布局的难题，CSS3新增了多列自动布局功能。

14.5.1 列宽度column-width

在CSS3中，可以使用column-width属性定义多列布局中每列的宽度，可以单独使用，也可以和其他多列布局属性组合使用。

column-width语法格式如下：

```
column-width: [<length> | auto]
```

其中属性值<length>是由浮点数和单位标识符组成的长度值，不可为负值；auto表示根据浏览器计算值自动设置。

【例14.7】实例文件：ch14\14.7.html

```
<html>
<head>
<title>多列布局属性</title>
<style>
body{
   -moz-column-width:300px;/*兼容Webkit引擎，指定列宽是300像素*/
   column-width:300px; /*CSS3标准化指定列宽是300像素*/
}
h1{
    color:#333333;
     background-color:#DCDCDC;
     padding:5px 8px;
     font-size:20px;
     text-align:center;
      padding:12px;
}
h2{
   font-size:16px;text-align:center;
}
p{color:#333333;font-size:14px;line-height:180%;text-indent:2em;}
</style>
</head>
<body>
<h1>春</h1>
<h2>朱自清</h2>
<p>
盼望着，盼望着，东风来了，春天的脚步近了。
</p><p>
一切都像刚睡醒的样子，欣欣然张开了眼。山朗润起来了，水涨起来了，太阳的脸红起来了。
</p><p>
小草偷偷地从土里钻出来，嫩嫩的，绿绿的。园子里，田野里，瞧去，一大片一大片满是的。坐着，躺着，打两个滚，踢几脚球，赛几趟跑，捉几回迷藏。风轻悄悄的，草软绵绵的。
</p>
…
</body>
</html>
```

在上面代码的body标记选择器中，使用column-width指定了要显示的多列布局的每列的宽度。下面分别定义标题h1、h3和段落p的样式，例如字体大小、字体颜色、行高和对齐方式等。

在Firefox5.0中浏览效果如图14-30所示，可以看到页面文章分为两列显示，列宽相同。

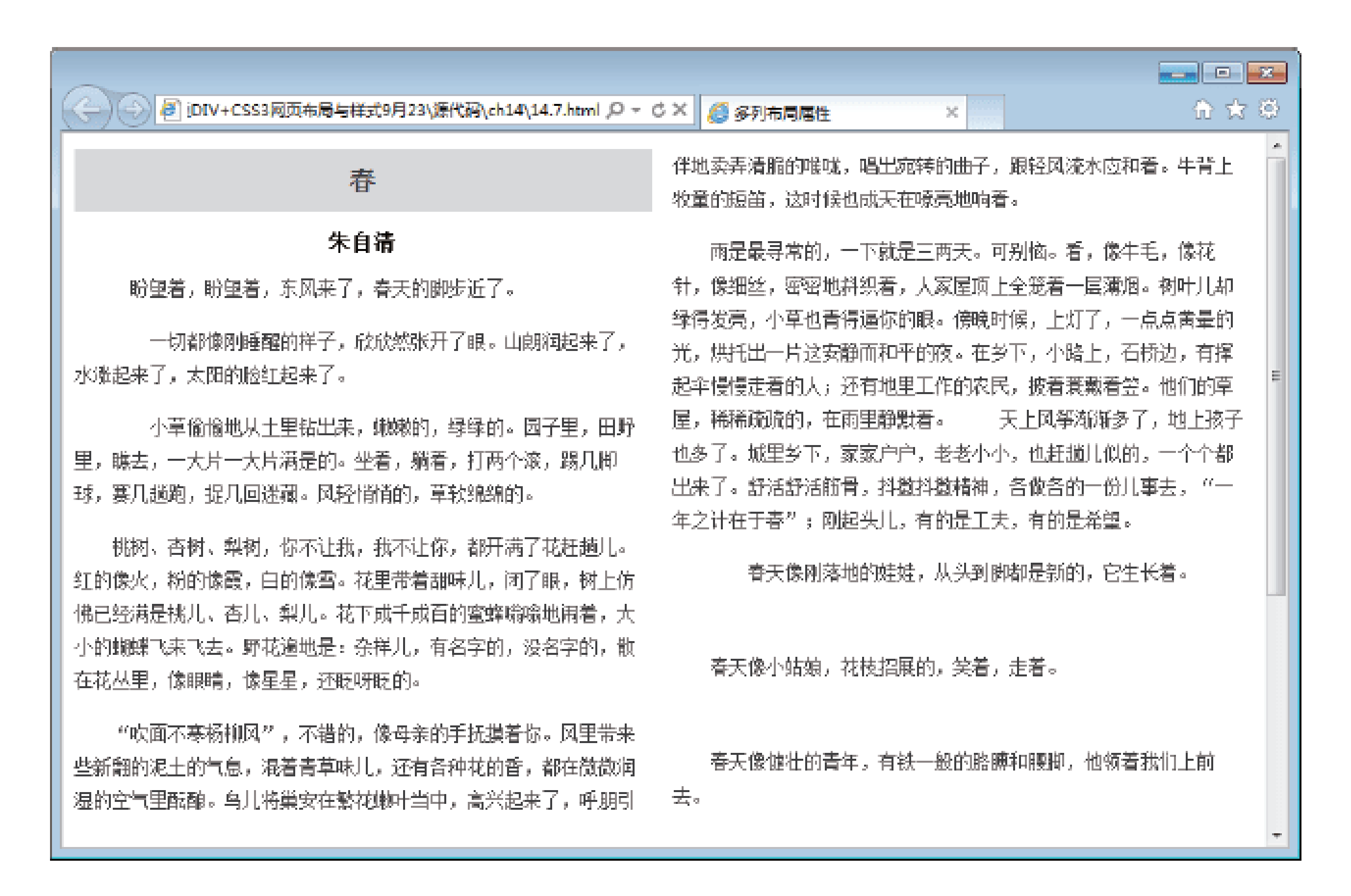


图14-30 设定列宽

14.5.2 列数column-count

在CSS3中，可以直接使用column-count指定多列布局的列数，而不需要通过列宽度自动调整列数。

column-count语法格式如下：

```
column-count: auto | <integer>
```

上面属性值<integer>表示值是一个整数，用于定义栏目的列数，取值为大于0的整数，不可以为负值；auto属性值表示根据浏览器计算值自动设置。

【例14.8】实例文件：ch14\14.8.html

```
<html>
<head>
<title>多列布局属性</title>
<style>
body{
   -moz-column-count:3;/*Webkit引擎定义多列布局列数*/
   column-count:3; /*CSS3标准定义多列布局列数*/
}
h1{
    color:#333333;
     background-color:#DCDCDC;
     padding:5px 8px;
     font-size:20px;
     text-align:center;
      padding:12px;
}
h2{
   font-size:16px;text-align:center;
}
p{color:#333333;font-size:14px;line-height:180%;text-indent:2em;}
</style>
```

```
    </head>
    <body>
    <h1>春</h1>
    <h2>朱自清</h2>
    <p>
      盼望着，盼望着，东风来了，春天的脚步近了。
    </p><p>
        一切都像刚睡醒的样子，欣欣然张开了眼。山朗润起来了，水涨起来了，太阳的脸红起来了。
    </p><p>
    小草偷偷地从土里钻出来，嫩嫩的，绿绿的。园子里，田野里，瞧去，一大片一大片满是的。坐着，
躺着，打两个滚，踢几脚球，赛几趟跑，捉几回迷藏。风轻悄悄的，草软绵绵的。
    </p>
    …
    </body>
    </html>
```

上面的CSS代码中，除了column-count属性设置外，其他样式属性和例14.7基本相同，就不介绍了。

在Firefox 5.0中浏览效果如图14-31所示，可以看到页面根据指定的情况显示了三列布局，其布局宽度由浏览器自动调整。

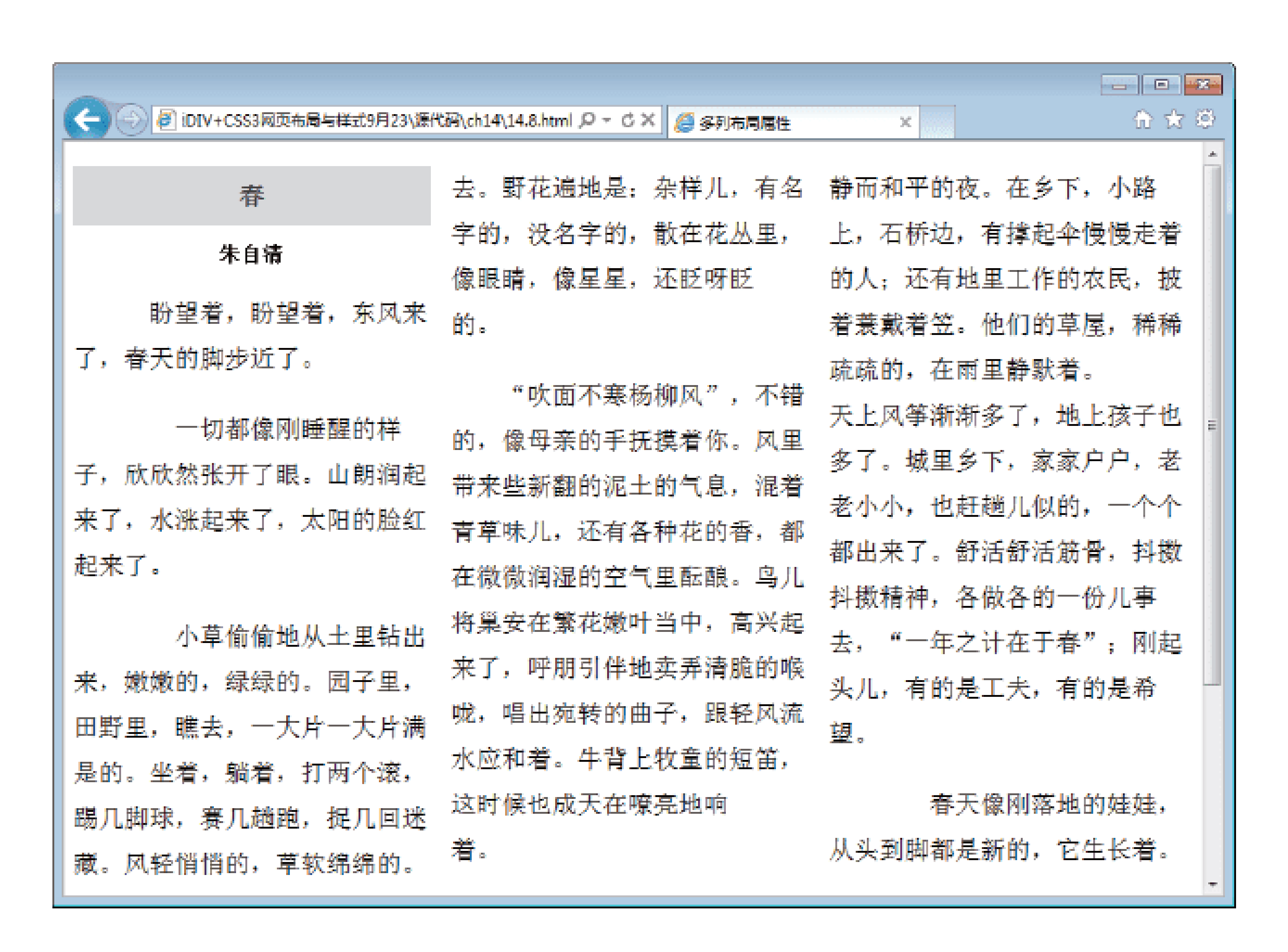


图14-31 设定列数

14.5.3 列间距column-gap

多列布局中，可以根据内容和喜好的不同，调整多列布局中列之间的距离，从而完成整体版式规划。在CSS3中，column-gap属性用于定义两列的间距。

column-gap语法格式如下：

```
column-gap: normal | <length>
```

其中属性值normal表示根据浏览器默认设置进行解析，一般为1em；属性值<length>表示值由浮点数和单位标识符组成的长度值，不可为负值。

【例14.9】实例文件：ch14\14.9.html

```
<html>
<head>
<title>多列布局属性</title>
<style>
body{
   -moz-column-count:3; /*Webkit引擎定义多列布局列数*/
   column-count:3; /*CSS3定义多列布局列数*/
   -moz-column-gap:3em; /*Webkit引擎定义多列布局列间距*/
   column-gap:3em; /*CSS3定义多列布局列间距*/
   line-height:2.5em;
}
h1{
    color:#333333;
     background-color:#DCDCDC;
     padding:5px 8px;
     font-size:20px;
     text-align:center;
      padding:12px;
}
h2{
   font-size:16px;text-align:center;
}
p{color:#333333;font-size:14px;line-height:180%;text-indent:2em;}
</style>
</head>
<body>
<h1>春</h1>
<h2>朱自清</h2>
<p>
  盼望着，盼望着，东风来了，春天的脚步近了。
</p><p>
    一切都像刚睡醒的样子，欣欣然张开了眼。山朗润起来了，水涨起来了，太阳的脸红起来了。
</p><p>
    小草偷偷地从土里钻出来，嫩嫩的，绿绿的。园子里，田野里，瞧去，一大片一大片满是的。坐着，躺着，打两个滚，踢几脚球，赛几趟跑，捉几回迷藏。风轻悄悄的，草软绵绵的。
</p>
…
</body>
</html>
```

上面代码中，使用-moz-column-count私有属性设定了多列布局的列数，-moz-column-gap私有属性设定列间距为3em，行高为2.5em。

在Firefox 5.0中浏览效果如图14-32所示，可以看到页面还是分为3个列，但列之间的距离相比较原来增大了不少。

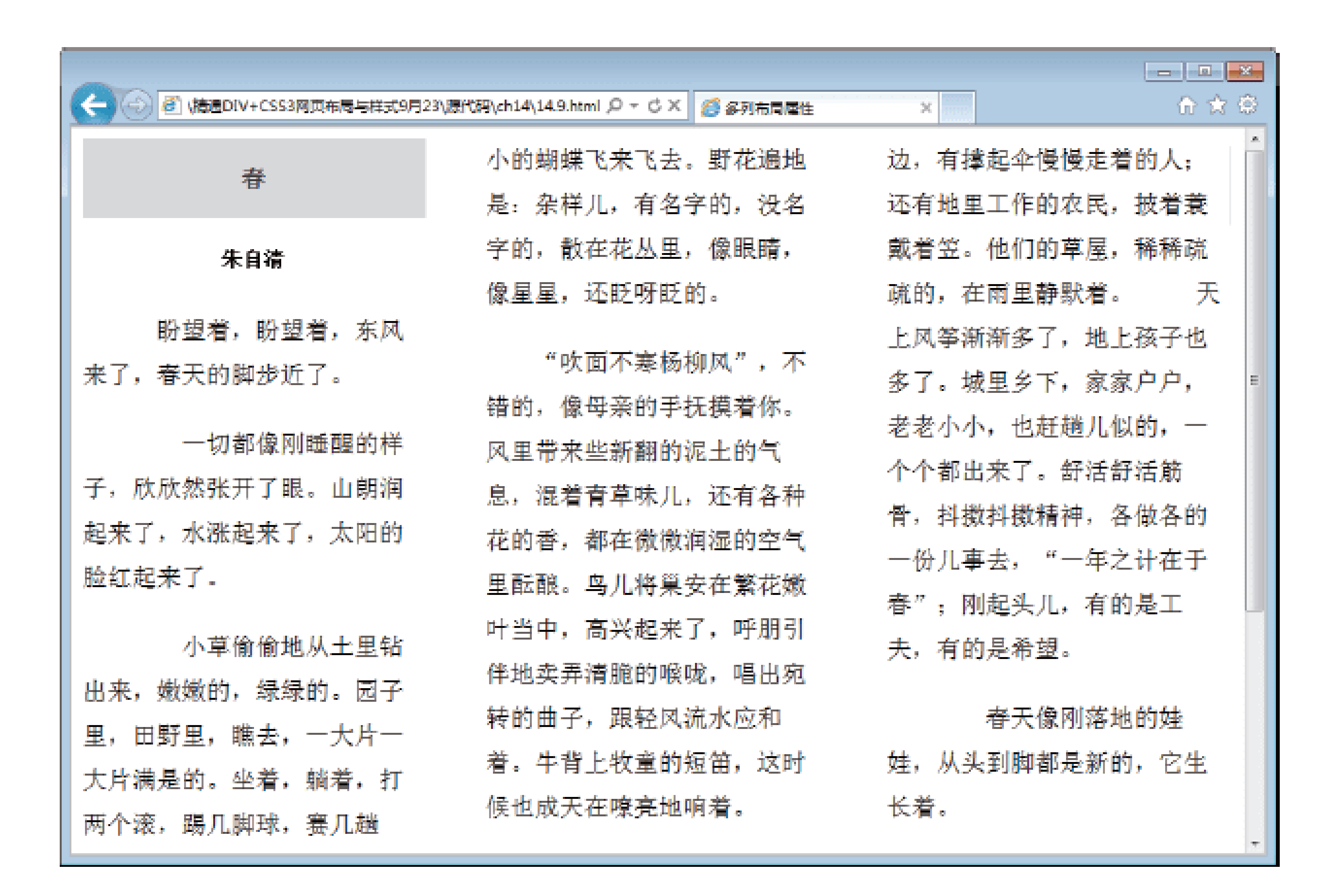


图14-32 设定列间距

14.5.4 列边框样式column-rule

边框是样式属性中不可缺少的属性之一，通过边框可以容易地界定边框内容，从而划分不同的区域。在多列布局中，同样可以设置多列布局的边框，用于区分不同的列数。在CSS3中，边框样式使用coumn-rule属性定义，包括边框宽度、边框颜色和边框样式等。

coumn-rule语法格式如下：

```
column-rule: <length> | <style> | <color>
```

其中属性值的含义如表14-1所示。

表14-1 column-rule属性值

属性值	说明
<length>	由浮点数和单位标识符组成的长度值，不可为负值。用于定义边框宽度，其功能和column-rule-width属性相同
<style>	定义边框样式，其功能和column-rule-style属性相同
<color>	定义边框颜色，其功能和column-rule-color属性相同

为了方便网页设计灵活的边框，CSS3在column-rule属性的基础上派生了3个列边框属性，分别为column-rule-width、column-rule-style和column-rule-color属性值。

column-rule-color属性用于定义边框颜色，其颜色值接受CSS3支持的所有颜色值。column-rule-width属性定义边框宽度，其属性值可以是任意浮点数。column-rule-style定义边框样式，其属性值和border-style属性值相同，即包括none、hidden、dotted、dashed、solid、double、groove、ridge、inset和ouset。

【例14.10】实例文件：ch14\14.10.html

```
<html>
<head>
<title>多列布局属性</title>
<style>
```

```
body{
    -moz-column-count:3;
    column-count:3;
    -moz-column-gap:3em;
    column-gap:3em;
    line-height:2.5em;
    -moz-column-rule:dashed 2px gray;/*Webkit引擎定义多列布局边框样式*/
    column-rule:dashed 2px gray; /*CSS3定义多列布局边框样式*/
}
h1{
     color:#333333;
      background-color:#DCDCDC;
      padding:5px 8px;
      font-size:20px;
      text-align:center;
       padding:12px;
}
h2{
    font-size:16px;text-align:center;
}
p{color:#333333;font-size:14px;line-height:180%;text-indent:2em;}
</style>
</head>
<body>
<h1>春</h1>
<h2>朱自清</h2>
<p>
  盼望着，盼望着，东风来了，春天的脚步近了。
</p><p>
    一切都像刚睡醒的样子，欣欣然张开了眼。山朗润起来了，水涨起来了，太阳的脸红起来了。
</p><p>
    小草偷偷地从土里钻出来，嫩嫩的，绿绿的。园子里，田野里，瞧去，一大片一大片满是的。坐
着，躺着，打两个滚，踢几脚球，赛几趟跑，捉几回迷藏。风轻悄悄的，草软绵绵的。
</p>
</body>
</html>
```

在body标记选择器中，定义了多列布局的列数、列间距和列边框样式，其边框样式是灰色破折线样式，宽度为2px。

在Firefox 5.0中浏览效果如图14-33所示，可以看到页面列之间添加了一个边框，其样式为破折线。

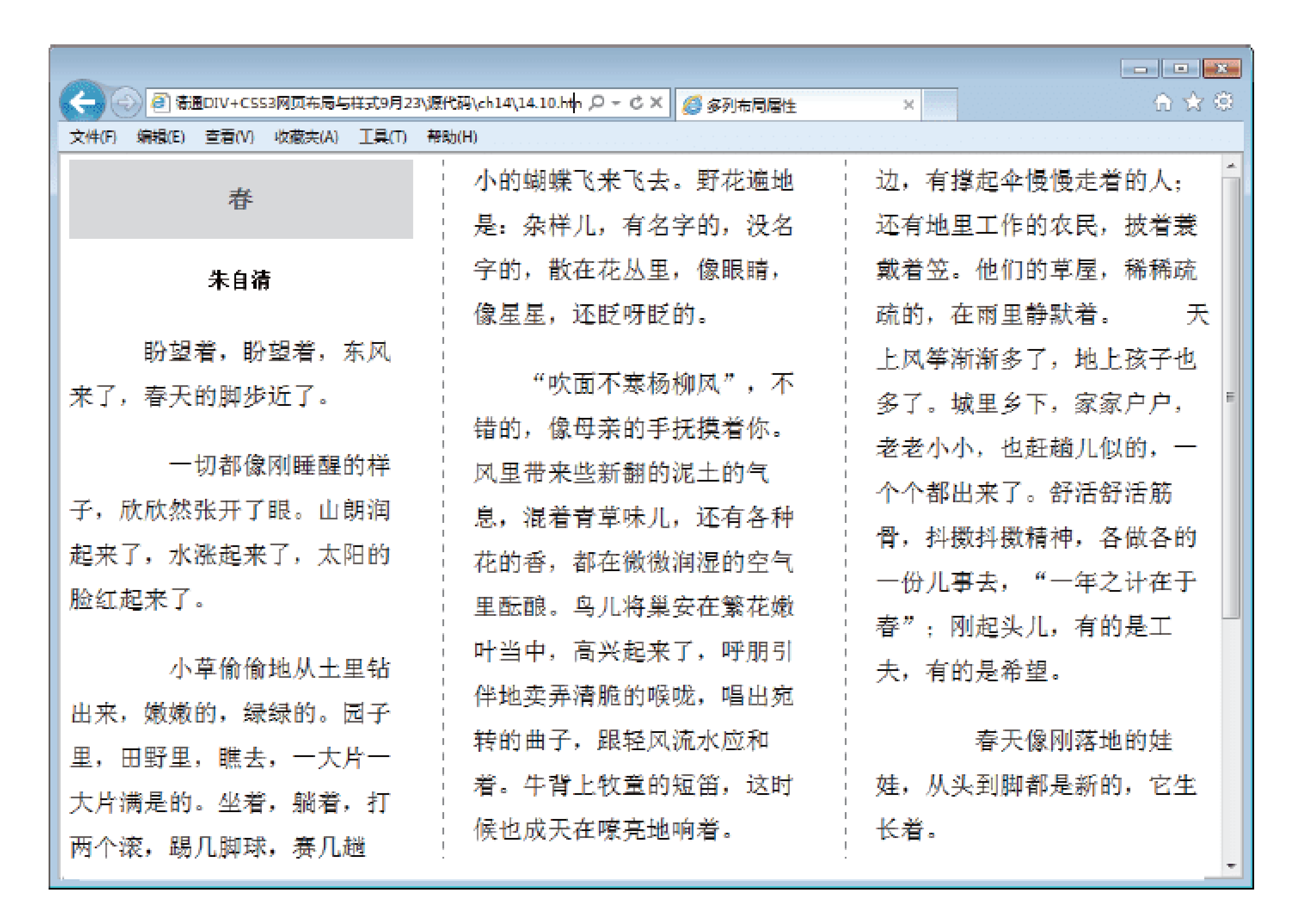


图14-33 定义边框样式



14.6 综合实例1——模拟起点中文网布局

在复杂的首页中，往往不是一种单一的结构，而是多种布局混合排版，其中多列排版布局是不错的选择。本实例模拟起点中文网首页中一个左中右排版的布局，其中左侧、右侧和中部3个DIV容器宽度固定，具体步骤如下所示。

01 分析需求。

创建一个左中右3列布局，现在有3种方式，一种是浮动布局，一种是定位布局，还有一种是CSS3的多列布局。这里使用了浮动布局，并固定左中右三侧宽度。实例完成后效果如图14-34所示。

02 创建HTML网页，使用DIV层分块。

从图14-34的效果可以看出，需要使用3个层分别设置左侧内容、中间内容和右侧内容，并将这3个DIV层都保护在一个DIV布局容器中，方便排版和调整。其代码如下：

```
<html>
<head>
<title>左中右版式</title>
</head>
<body>
<div class="big">
  <div class="left">
   <h3>月票PK榜</h3>
   <table border=0 width=178px>
    <tr><td align=left>我的老婆是军阀</td><td align=right>719</td></tr>
```

```
        <tr><td align=left>仙绝</td><td align=right>329</td></tr>
        <tr><td align=left>大魔神</td><td align=right>322</td></tr>
        <tr><td align=left>最强掌门</td><td align=right>253</td></tr>
        <tr><td align=left>术士的幸福生活</td><td align=right>225</td></tr>
        <tr><td align=left>超级电能</td><td align=right> 223</td></tr>
        <tr><td align=left>超级警官</td><td align=right> 203</td></tr>
        <tr><td colspan=2 align=right>更多>></td></tr>
       </table>
      </div>
      <div class="middle">
      <h3>图书专题</h3>
      <p class=title>离婚妈妈的快乐情缘  如果不能好好爱</p>      <p class=content>故
事的主角叫许可。30岁，失业，离婚，还带个孩子,一夜之间，世界上所有倒霉的事情都找上门来...<a
href="">[更多内容]</a>
    </p>
    <p class=title>著名作家陈丹燕的心灵旅程我要游过大海</p>
    <p class=content>游历世界各地，然后记叙成书，陈丹燕爱尔兰旅行散集《我要游过大海》，讲
述爱尔兰的传奇。<a href="">[更多内容]</a></p>
    </div>
      <div class="right">
    <img src=98.jpg >
    <p>阴差阳错，陈默然回到了八国联军那会……
      </div>
    </div>
    </body>
    </html>
```

上面代码定义了big层作为布局容器，left层作为左侧容器，middle层作为中间容器，right层作为右侧容器。

在IE9.0中浏览效果如图14-35所示，可以看到页面信息从上到下自动排列，没有任何修饰样式。

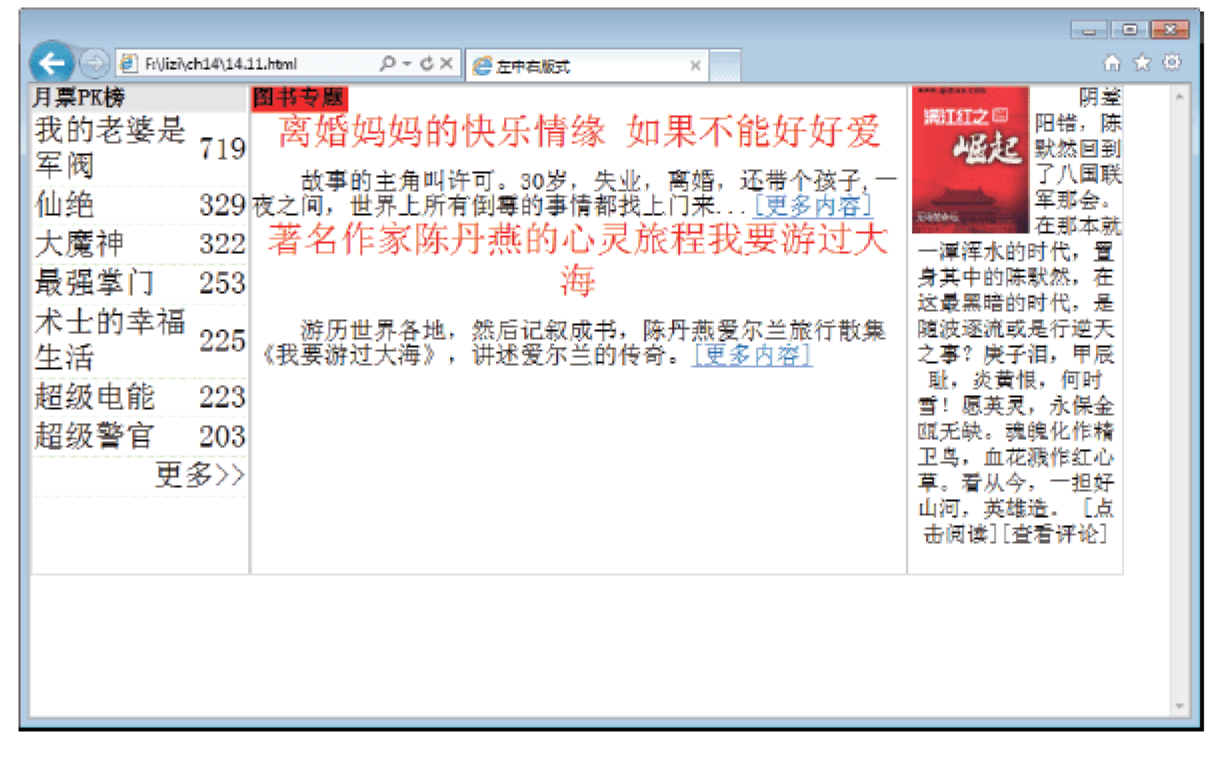


图14-34　左中右三列布局

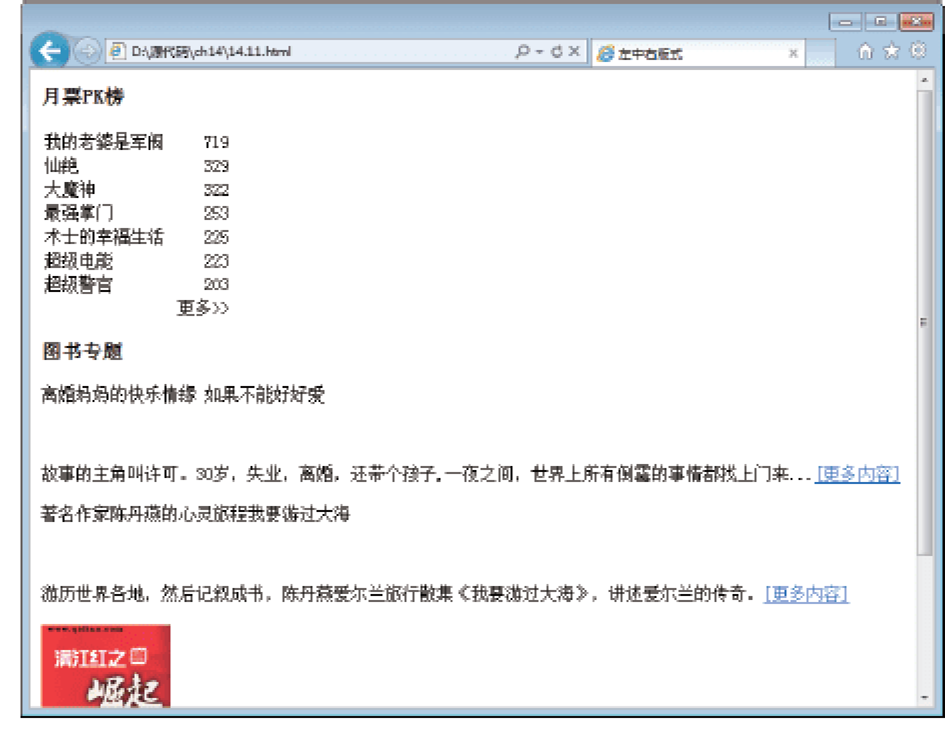


图14-35　DIV层划分块

03 添加CSS代码，定义全局样式。

```
<style>
* {
  padding:0px;
  margin:0px;
}
body {
  font-family:"宋体";
  font-size:12px;
}
.big {
  width:900px;
  margin:0 auto 0 auto;
  height:300px;
}
</style>
```

上面代码定义了全局的显示样式，例如body标记选择器中定义了字形为“宋体”，文字大小为12px。在类选择器big中定义了布局容器宽度为900px，高度为300px，上下外边距为0，左右外边距自动调整。

在IE9.0中浏览效果如图14-36所示，可以看到字体变小，网页在一定范围内显示。

04 添加CSS代码，定义左侧样式。

```
.left {
  width:178px;
  float:left;
  height:200px;
            border:1px solid #CBCBCB;
}
.left h3{
     background-color:E9E9E9;
     text-align:left;
     font-weight:bolder;
}
.left table td{
            list-style-type:none;
            font-size:15px;
            border-bottom:1px dotted #ddeecc;
            color:#234628;
            font-family:"宋体";
   }
```

上面代码中，选择器left定义了层宽度为178px，高度为200px，在网页左边浮动布局。下面分别定义标题h3和表格单元格的显示样式，例如对齐方式、文字大小、边框样式等。

在IE9.0中浏览效果如图14-37所示，可以看到页面左侧以绿色字体显示，并且每行文本下面带绿色边框。

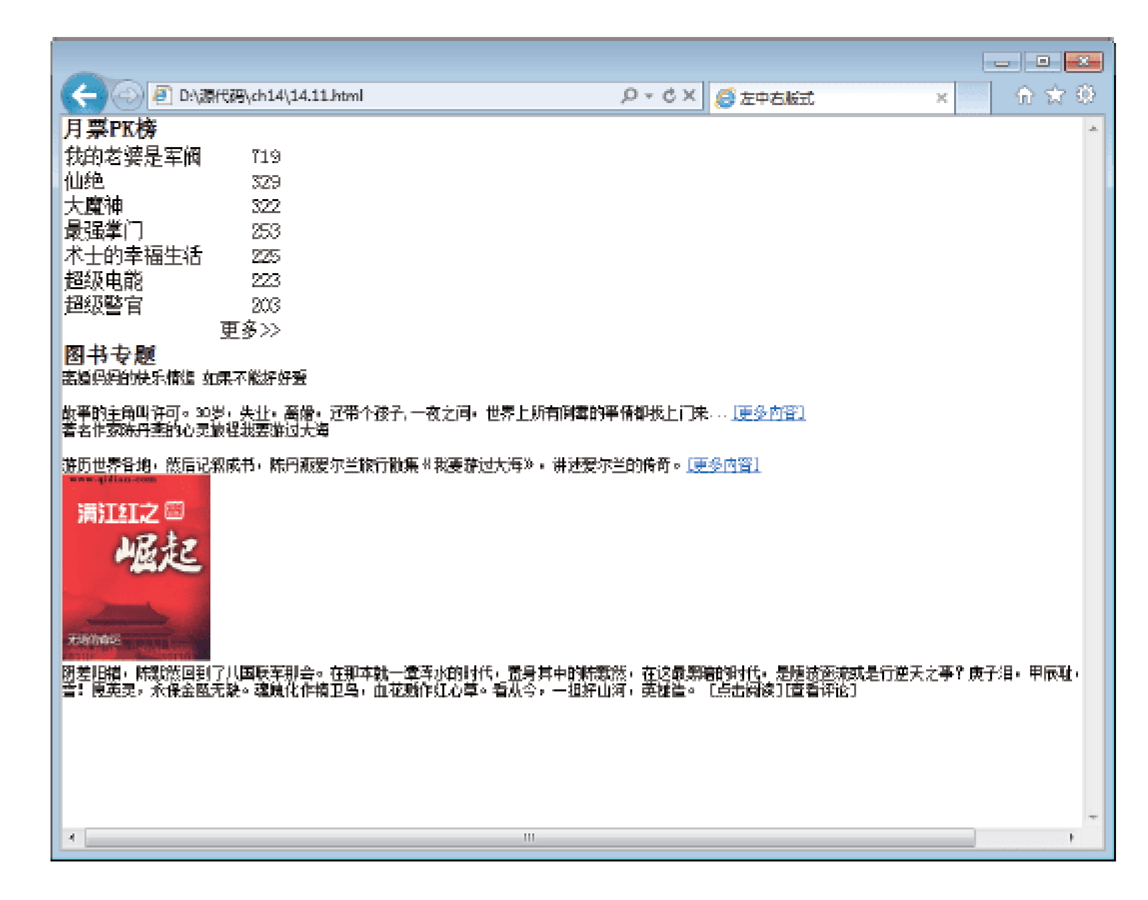


图14-36 定义全局样式



图14-37 定义左侧样式

05 添加CSS代码，定义中间样式。

```
.middle {
  width:538px;
  float:left;
  height:200px;
   border:1px solid #CBCBCB;
}
.middle h3{
        background-color:red;
         width:80px;
}
.title{
    font-size:16px;
    color:red;
    text-align:center;
}
.content{
    text-indent:2em;
    line-height:12px;
    font-size:13px;
}
```

上面代码中，选择器middle定义了中间宽度为538px，高度为200px，边框样式和浮动布局。其他选择器对背景色、对齐方式、颜色、行高和缩减进行了定义。

在IE9.0中浏览效果如图14-38所示，可以看到中间部分带有边框显示，边框中以红色显示标题信息，段首缩减两个文字。

06 添加CSS代码，定义右侧样式。

```
.right {
  width:178px;
  float:left;
  height:200px;
  text-align:center;
```

```
        border:1px solid #CBCBCB;
    }
    img{
           float:left;
           max-height:80px;

    }
    .right p{
          font-size:10px;
          text-indent:2em;
    }
```

上面right选择器定义右侧容器宽度为178px、高度为200px、居中对齐、浮动在左侧显示和边框样式。下面选择器对图片和段落进行样式定义。

在IE9.0中浏览效果如图14-34所示，可以看到页面3个矩形框并列在页面显示，右侧显示了一张图片和段落，文本信息环绕图片显示。

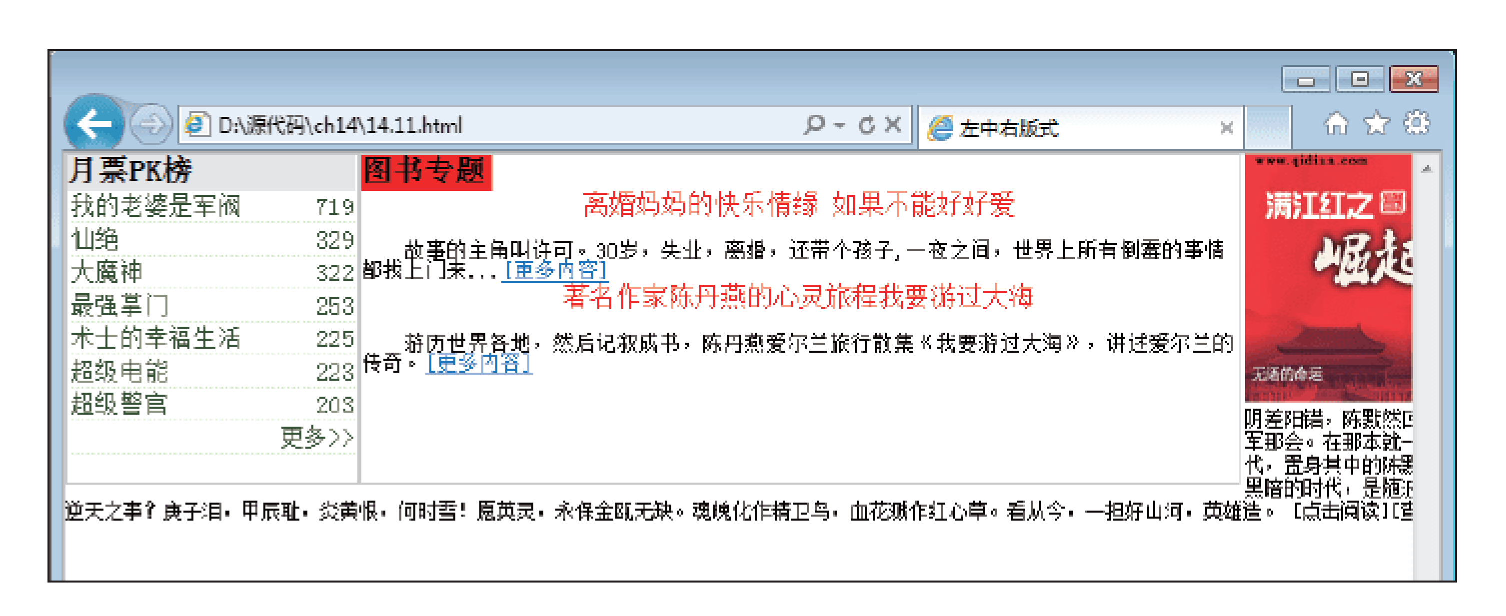


图14-38 定义中间样式



14.7 综合实例2——电子企业首页

小型企业网站往往非常简单，导航信息不多但要求突出主题。本实例模拟一个电子企业首页，总体上是上中下结构，中间采取左右版式，具体步骤如下所示。

01 分析需求。

创建一个一个上中下的布局，再加上中间的左右版式，需要5个DIV层。实例完成后，页面效果如图14-39所示。

02 创建HTML网页，使用DIV层分块。

```
    <html>
    <head>
    <title>两列布局</title>
    </head>
    <body>
    <div class="big">
```

```
        <div class="nav">
          <ul>
              <li><a href="#">首页</a></li>
                <li><a href="#">企业新闻</a></li>
                <li><a href="#">视听新闻</a></li>
                <li><a href="#">市场快讯</a></li>
                 <li><a href="#">投资导引</a></li>
                  <li><a href="#">每天报道</a></li>
            </ul>
        </div>
        <div class="main">
          <div class="left">
               <div class="wen">
            <h1>中国电子首次入选《财富》世界500强 </h1>
              <p>7月7日，美国《财富》杂志发布2011年度世界500强企业排名，中国电子信息产业
集团有限公司首次入选世界500强企业……
    </p>
            </div>
            </div>
            <div class="right">
                <img src="b0.jpg" border="0" />
                <p>《精算商业管理系统》针对中小企业的特点，集进销存财务管理一体化。帮助企
业全面管理商品帐、资金帐、往来帐、费用帐……
    </p>
            </div>
        </div>
        <div class="foot">
          <p>2011 郑州水月网页工作室 版权所有</p>
        </div>
    </div>
    </body>
    </html>
```

上面代码中，big层是一个布局容器，用来存放其他DIV层；nav层用来作为页头部分；main层是网页主体，包含两个DIV层；left层为页面主体左侧；right层为页面主体右侧；foot层表示页脚部分。

在IE9.0中浏览效果如图14-40所示，可以看到自上而下排列，上面是超级链接信息，下面是文本和图片。



图14-39 电子企业首页

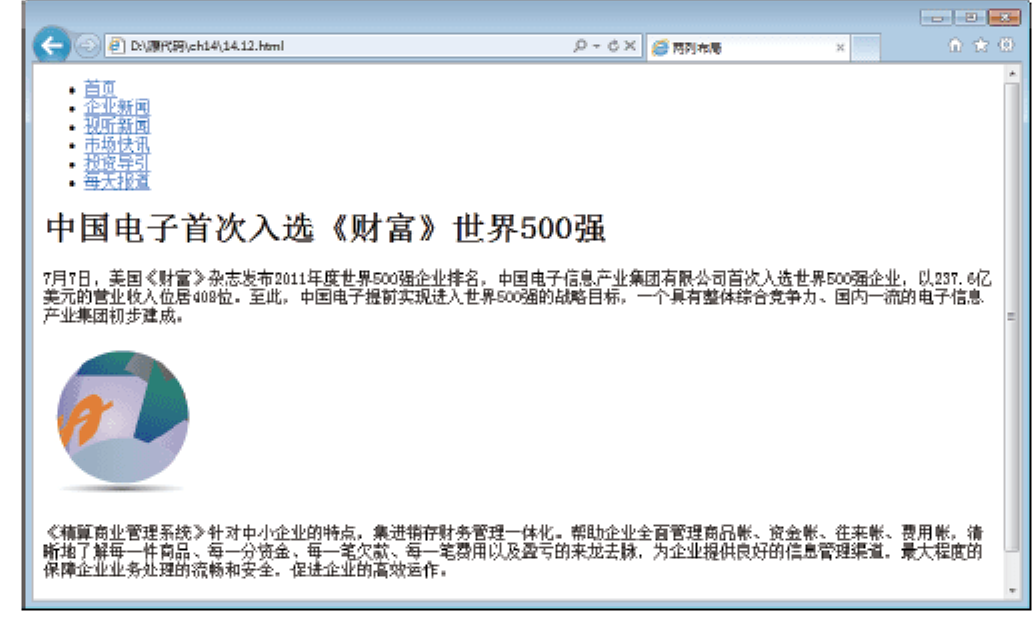


图14-40 基本HTML页面

03 添加CSS修饰整体样式。

```
*{
padding:0px;
margin:0px;
}
body{
  font-family:"宋体";
  font-size:12px;
  }
.big{
  width:900px;
  margin:0 auto 0 auto;
  }
```

上面代码定义了全局样式，例如网页上字体为宋体，文字大小为12px，内外边距都是0px。在big选择器中，定义整个布局容器宽度为900px。

在IE9.0中浏览效果如图14-41所示，可以看到页面字体变小并以宋体显示。

04 添加CSS，修饰页头部分。

```
.nav{
  background-color:#C2E3E9;
  width:800px;
  height:80px;
   background-image:url(80.jpg);
  }
.nav ul{
  list-style-type:none;
  }
.nav li{
  float:left;
  }
.nav a{
  display:block;
  width:120px;
  height:20px;
  line-height:20px;
  text-align:center;
  color:#6600CC;
  text-decoration:none;
  }
.nav a:hover{
  width:120px;
  height:20px;
  color:#990000;
  background-color:#CCCCCC;
  }
```

在选择器nav中，定义了背景色、背景图片、宽度和高度。下面的选择器分别定义了导航菜单的显示样式，例如无序列表不显示符号、字体大小、颜色、对齐方式和行高等。

在IE9.0中浏览效果如图14-42所示，可以看到页面顶部显示了一个图片，其导航菜单在

图片顶部显示。



图14-41 定义全局样式

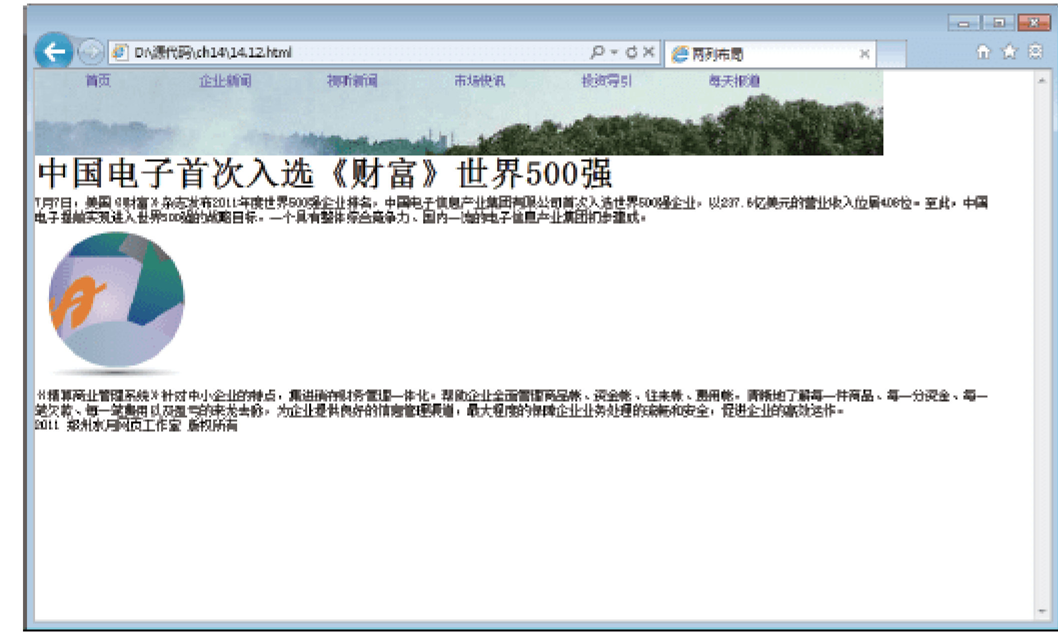


图14-42 定义页头部分

05 添加CSS，修饰页面主体。

```
.main{
    width:800px;
    height:400px;
    }
  .left{
    width:618px;
    float:left;
    height:400px;
    text-align:center;
     border:1px solid #cbcbcb;
    }
  .left img{
    margin:10px 0 10px 0;
    }
.left p{
                          text-indent:2em;
                          text-align:left;
                          font-size:13px;
                          line-height:30px;
                 }
  .right{
    width:178px;
    float:left;
    height:400px;
    text-align:center;
    border:1px solid #cbcbcb;
    }
.right p{
                    text-indent:2em;
                   font-size:13px;
                  }
```

上面代码中，main选择器定义中间容器整体宽度为800px，高度为400px；left选择定义网页主体左侧部分样式，例如宽度为618px，高度为400px，float指定div层在左侧浮动布局；right选择器定义页面主体右侧部分，例如宽度为178px、高度为400px、文本居中对齐等。

在IE9.0中浏览效果如图14-43所示，可以看到页面中间部分分为两个边框显示，左侧部分显示了段落信息，右侧部分显示了图片和文本信息。

06 添加CSS，修饰页脚部分。

```
.foot{
  height:15px;
  text-align:center;
  }
 .foot p{
                  font-style:italic;
                  font-size:10px;
                  width:800px;
                   text-align:center;
    }
```

上面代码定义了页脚部分显示样式，例如页脚高度为15px、文本居中对齐、段落样式中文字大小为10px、居中对齐等。

在IE9.0中浏览效果如图14-39所示，可以看到页脚部分居中显示，其字体以斜体显示。



图14-43 定义主体部分



14.8 综合实例3——汽车网站首页

汽车网站是现在网站的主流之一，一般用于宣传和展示新车信息。本实例模拟一个汽车网站首页，具体步骤如下所示。

01 分析需求。

创建一个上中下结构网页，如果宽度固定就非常容易了。这里需要创建4个div层，一个用来作为布局容器，另外3个分别对应上中下三块。实例完成后效果如图14-44所示。

02 创建HTML，使用div层分块。

```
    <html>
    <head>
    <title>汽车首页</title>
    </head>
    <body>
    <div class="big">
      <div class="logo">
      <img src="26.jpg" />
      </div>
        <div class="up">
        <br />
          <h1>深圳三维比亚迪  荣誉呈现F6大运尊享版
    </h1>
          <p> 浓情七月，魅力八月……
     </p>
          <p>  F6作为比亚迪的首款中高级战略车型…… </p>
        </div>
            <div class="foot">
            <br />
              <p><a href="#">首页</a> | <a href="#">论坛</a> | <a href="#">图库
</a> | <a href="#">二手车</a> | <a href="#">团购</a></p>
                <p>2011 &copy; 郑州水月工作室  技术支持</p>
            </div>
    </div>
    </body>
    </html>
```

上面代码中，定义了4个层，其中big层作为布局容器，logo层作为页头部分，up层作为页面主体，foot层页脚部分。

在IE9.0中浏览效果如图14-45所示，可以看到页面从上到下显示3个部分，包括图片、段落和导航菜单。



图14-44　汽车网站首页



图14-45　基本HTML页面

03 添加CSS样式，定义全局样式。

```
    <style>
    * {
      padding:0px;
```

```
    margin:0px;
  }
  body {
    font-family:"宋体";
    font-size:12px;
  }
  .big {
    width:800px;
    margin:0 auto 0 auto;
  }
  </style>
```

上面代码中，body选择器中定义了页面字体为宋体，字体大小是12px，big选择器中定义了布局容器宽度为800px和外边距等。

在IE9.0中浏览效果如图14-46所示，可以看到图片和文字居中显示，并且字体大小发生变化。

04 添加CSS样式，修饰页头部分。

```
  .logo {
    width:800px;
    height:100px;
  }
```

上面代码中，选择器logo定义页头部分，宽度为800px，高度为100px。此页面只是图片高度发生变化，这里就不再显示了。

05 添加CSS样式，修饰中间样式。

```
  .up{
    width:800px;
    margin:5px auto 20px auto;
                border:#0066FF 1px solid;
    }
  .up h1{
    font-size:18px;
    text-align:center;
    margin-bottom:10px;
    }
  .up p{
              text-indent:2em;
  }
```

上面代码中，up选择器定义了页面主体宽度、边框样式和外边距等。下面选择器定义了标题颜色。

在IE9.0中浏览效果如图14-47所示，可以看到页面主体部分以边框形式显示，段首缩进两个字符显示。



图14-46 定义全局样式



图14-47 定义页面主体样式

06 添加CSS样式，修饰页脚部分。

```
.foot{
  width:800px;
  height:80px;
  text-align:center;
  margin-top:5px;
  }
.foot p{
  margin-bottom:5px;
              font-style:italic;
  }
a{
    line-height:20px;
    color:#000000;
    text-decoration:none;
    text-align:center;
    }
a:hover{
    color:#0000FF;
    text-decoration:underline;
    }
```

上面代码中，foot选择器定义了页脚部分宽度为800px，高度为80px，文本居中对齐，上外边距为5px。.foot p和a:hover选择器分别定义了段落、超级链接的显示样式。

在IE9.0中浏览效果如图14-44所示，可以看到页脚部分显示了一个简单的导航菜单和版权信息，文字以斜体显示。

14.9 专家解惑

1．IE浏览器和Firefox浏览器，显示float浮动布局会出现不同的效果，为什么？

两个相连的DIV块，如果一个设置为左浮动，一个设置为右浮动，这时在Firefox浏览器中就会出现设置失效的问题。其原因是IE浏览器会根据设置来判断float浮动，而在Firefox中，如果上一个float没有被清除的话，下一个float会自动沿用上一个float的设置，而不使用自己的float设置。

这个问题的解决办法就是，在每一个DIV块设置float后，在最后加入一句清除浮动的代码clear:both，这样就会清除前一个浮动的设置了，下一个float也就不会再使用上一个浮动设置，从而使用自己所设置的浮动了。

2．DIV层高度是设置好还是不设置好？

在IE浏览器中，如果设置了高度值，但是内容很多，会超出所设置的高度，这时浏览器就会自己撑开高度，以达到显示全部内容的效果，不受所设置的高度值限制。而在Firefox浏览器中，如果固定了高度的值，那么容器的高度就会被固定住，就算内容过多，DIV也不会撑开，也会显示全部内容，但是如果容器下面还有内容的话，那么这一块就会与下一块内容重合。

这个问题的解决办法就是，不设置高度的值，这样浏览器就会根据内容自动判断高度，也就不会出现内容重合的问题。

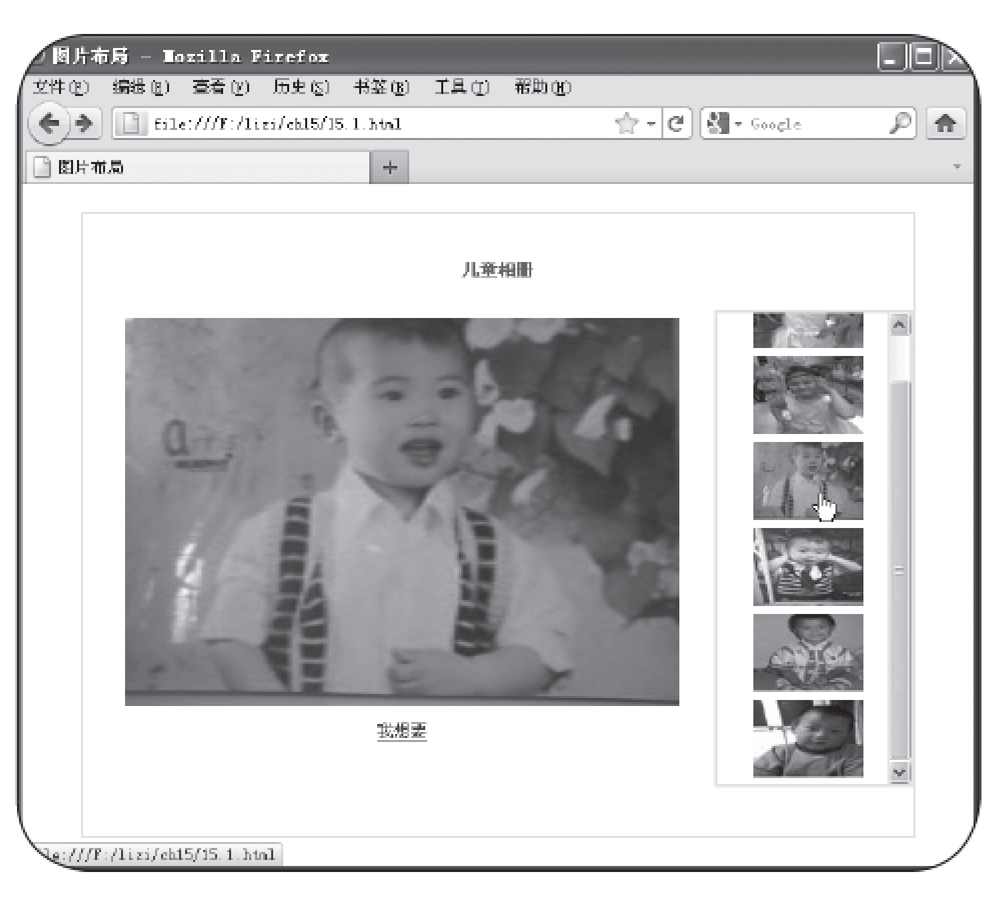


第 15 章

DIV+CSS3 布局实战

本章引言

DIV+CSS是一种网页布局方法，这种网页布局方法有别于传统的Table布局，真正地达到了W3C提倡的内容与表现相分离。本章将进一步介绍利用CSS3样式进行DIV+CSS网页布局的知识。







15.1 DIV布局优势

随着Web标准化设计理念的普及，国内很多大型门户网站已经纷纷采用DIV+CSS制作方法，DIV+CSS布局是网页设计中的一种布局方式，和HTML网页设计语言中的表格（Table）定位布局作用类似。在网站新设计标准中，不再使用表格定位技术，而是采用DIV+CSS的方式实现各种定位。

使用DIV+CSS布局，有以下优势：

(1) 符合W3C标准。这保证网站不会因为将来网络应用的升级而被淘汰。

(2) 大大缩减页面代码，提高页面浏览速度，缩减带宽成本。

(3) 样式调整更加方便。内容和样式的分离，使页面和样式的调整变得更加方便。在团队开发中更容易分工合作而减少相互关联性。

(4) 轻松控制页面布局，方便维护，可以将许多网页的风格样式同时更新，不用再一页一页地更新了。可以将站点上所有的网页风格统一使用一个CSS文件进行控制，只要修改这个CSS文件中相应的行，整个站点的所有页面的效果都会随之发生变化。

(5) 搜索引擎更加友好。相对于传统的Table，采用DIV+CSS技术的网页，对于搜索引擎的收录更加友好，结构更加清晰，容易被搜索引擎搜索到，优化了SEO。



15.2 综合实例1——家庭相册

本节将使用CSS样式创建一个常用的家庭相册。由于目前IE浏览器对CSS3标准支持力度不够，故此相册不能在IE浏览器显示，具体步骤如下所示。

01 分析需求。

本实例模拟百度图片中图片的显示样式，分为显示图片区域和图片列表区域，即布局为左右版式。实例完成后，效果如图15-1所示。

02 构建HTML页面，使用DIV搭建框架。

```
<html>
<head>
<title>图片布局</title>
</head>
<body>
<div id="info">
  <h3>儿童相册</h3>
  <div id="zs">
```

```
        <ul>
          <li><a href="" class="b1 z" ><span><img src="a1.jpg" /><br />
            ok，可以拍了!<br />
           </span></a></li>
          <li><a href="" class="b2 z"><span><img src="a2.jpg"/><br />
            听，这是什么声音呀？<br />
           </span></a></li>
          <li><a href="" class="b3 z"><span><img src="a3.jpg" /><br />
            我想要<br />
           </span></a></li>
          <li><a href="" class="b4 z"><span><img src="a4.jpg" /><br />
            放炮了，哈，<br />
           </span></a></li>
          <li><a href="" class="b5 z"><span><img src="a5.jpg"  /><br />
            我淑女吧<br />
           </span></a></li>
          <li><a href="" class="b6 z"><span><img src="a6.jpg"  /><br />
            为什么天变黑了<br />
           </span></a></li>
        </ul>
      </div>
    </div>
    </body>
    </html>
```

上面代码中创建了两个层，分别为info层和zs层。其中info是一个布局容器，里面是其他的HTML元素，例如标题、图片等。zs层主要存放超级链接，即无序列表等。

在Firefox 5.0中，浏览效果如图15-2所示，可以看到页面左侧图片，根据无序列表自上而下进行排列，每张图片下都有说明文字。



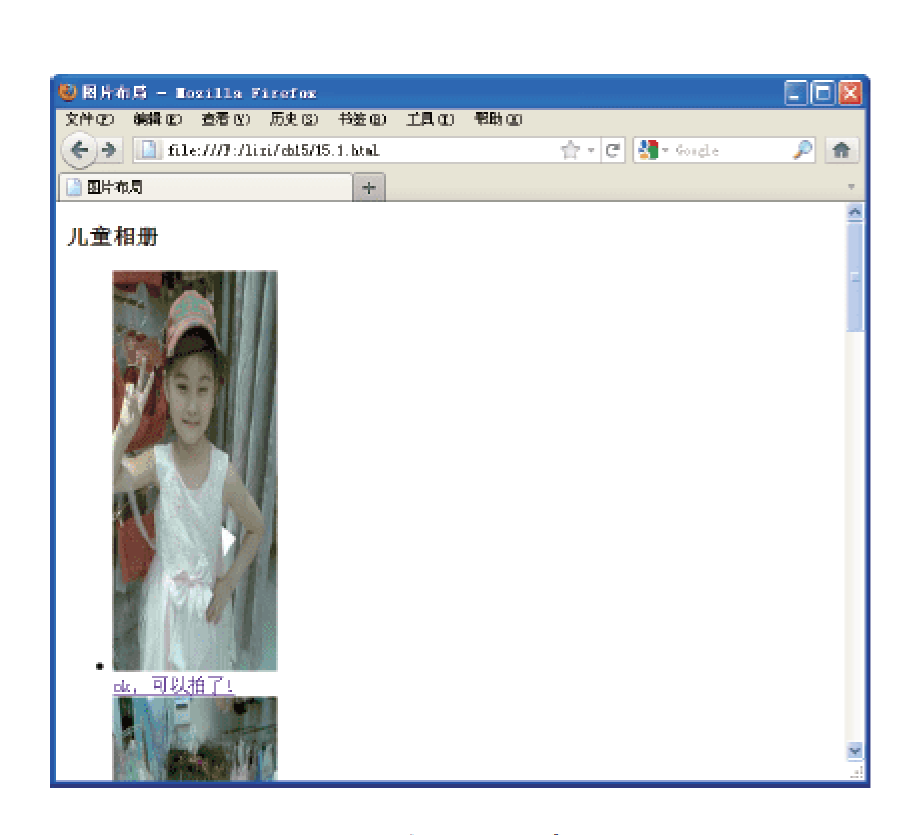




图15-1 相册左右版式　　　图15-2 创建基本HTML页面

03 添加CSS样式，对全局修饰。

```
    <style>
    body{
     margin:0;
     padding:0;
```

```
      font-size:12px;
      line-height:1.7;
      font-family:Verdana, "宋体";
      overflow:hidden
    }
    #info{
     width:600px;
     margin-left:auto;
     margin-right:auto;
      text-align:center;
      border:1px solid #ddeecc;
       height:450px;
       margin-top:20px;
    }
    #zs{
     height:340px;
    overflow: auto;
    float:right;
     width:140px;
    margin-top:20px;
    margin-bottom:50px;
    border:2px solid #ddeecc;
    }
    </style>
```

上面代码中，创建了3个ID选择器，其中body用于定义网页全局样式，例如内外边距、文字大小、行高、字体和隐藏等。info选择器用于定义布局容器宽度、高度、外边距上左右两侧距离、对齐方式、边框等。zs选择器用于定义无序列表的高度、宽度、浮动布局、边框和外边距等。

在Firefox 5.0中，浏览效果如图15-3所示，可以看到网页中显示了一个大的边框，边框中包含标题和图片信息，其中列表图片在右侧显示，并在指定边框显示，标题居中显示。此时鼠标放到图片上，没有任何情况发生。

04 添加修饰样式，修饰局部样式。

```
    h3{
    margin:30px 0 0 0;
    color:#CC0000;
     font-size:12px;
    }
    span{
    visibility:hidden;
    position:absolute;
    overflow:hidden;
    }
    ul,li{
     list-style:none;
    margin:0;
    padding:0;}
    a:active,a:hover{
```

```
cursor:pointer
}
a:hover span,a:active span{
position:absolute;
 top:90px;
z-index:20px;
visibility: visible;
 margin-left:-500px;
}
#info img{
width:400px;
height:280px;
border:7px solid #FFFFFF
}
.z{
width:80px;
height:56px;
display:block;
border:1px solid #FFFFFF;
margin:4px 0 4px 25px;
}
.b1{ background:url(b1.jpg)}
.b2{ background:url(b2.jpg)}
.b3{ background:url(b3.jpg)}
.b4{ background:url(b4.jpg)}
.b5{ background:url(b5.jpg)}
.b6{ background:url(b6.jpg)}
```

上面代码中，创建了几个选择器，其中h3、span、ul、li和a:hover等修饰对应的HTML标记样式，下面的类选择器z用于定义超级链接的显示样式，例如宽度、高度、块显示、边框和外边距等。b1设定背景图片，即鼠标放到第一个无序列表选项时，就会显示此背景图片。b2等以此类推。

在Firefox 5.0中，浏览效果如图15-4所示，可以看到图片在网页左侧显示，右侧显示了一个列表框。当鼠标放到列表中的任一图片上，会在网页左侧显示该图片。



图15-3 设置全局样式



图15-4 最终效果

15.3 综合实例2——个人网站首页

为了表达观点、抒发感情或者宣传自己，个人网站是很好的途径之一。本实例创建一个个人网站，具体步骤如下所示。

01 分析需求。

首先需要使用div层将页面划分不同的区域，用以构建网页布局。并创建一个DIV层作为布局容器，其他div层分别代表不同的区域。实例完成后效果如图15-5所示。

02 创建HTML网页，使用div层分块。

在HTML页面，根据图15-5所示的页面效果，总体可以划分为上中下结构，网页主体内容分为左中右版式。其代码如下：

```
<html>
<head>
<title>朴实的一天</title>
</head>
<body>
<div id="wrapper">
<div id="header"> <h1> 李四  <span class="orange"> 的网站 </span></h1>
</div>
<div id="navhorisontell">
<ul>
<li><a href="index.htm" class="selected">博客首页 </a> </li>
<li><a href="index.htm">关于李四 </a></li>
<li><a href="index.htm">给我留言 </a></li>
<li><a href="index.htm">友情链接 </a></li>
<li><a href="index.htm">我的信箱 </a></li>
<li><a href="index.htm">社区 </a></li>
</ul>
</div>
<div id="content">
<div id="col1">
<h1>2011.08.12 晚上</h1>
<p> 好热啊，早上起来脑袋都昏沉，直到现在都是。 机器还在响着，车间里更热，所以我不去开空调，只希望自然风再大些。 那些可爱的人们在我的眼皮底下辛苦的干着活，又脏又累，我怎么忍心？？？把褥子被子都拿出去晒了，太阳出来又进去，到底是怎么啦提不起劲儿？ 这些工作都做完，喝了一杯温开水，汗流浃背，衣服湿透了，脑袋依然昏沉。 宿舍里很压抑，不知道是自己的原因还是环境就这样？
</p>
<p>当年张爸爸在工地上，没有风扇，没有空调，只要一把旧芭蕉扇就度过了几个夏天，何况我晚上还有一台电风扇，比起他的艰辛我这又算什么。 张妈妈在家里放了关于佛的书籍还有音乐，每天重复着大慈大悲，这些音乐听得我特别感动，一有空闲就想听听这样的音乐看看这些书。 他们本来就是一群超善良的人们，过于懂得感恩是值得善待的人们，想到他们我就特心疼特忘情。
</p>
<p>昨天感冒了，这种天气下还能感冒，晚上吃了药今天就好了，好神奇啊，身体不舒服一剂药就可以治愈，如果心不舒服了该怎么治疗？也许只是一句问候，一声关怀。可是，就是因为冷漠，才让人更自
```

```
立的吧！
    </p>
    </div>
    <div id="col2"><img src="Img/bild.jpg" class="imagecol2" alt="bild"></img>
</div>
    </div>
    <div id="nav">
    <ul>
    <li><a href="index.htm" class="selected">日志 </a> </li>
    <li><a href="index.htm">相册 </a></li>
    <li><a href="index.htm">个人档案 </a></li>
    <li><a href="index.htm">分享 </a></li>
    <li><a href="index.htm">音乐地带 </a></li>
    <li><a href="index.htm">更多 </a></li>
    </ul>
    </div>
    <div id="footer"><p>2011 （网站） 设计者 <a href="">李四工作室  </a> </p></div>
    </div>
    </body>
    </html>
```

上面代码中，wrapper层用做布局容器，放置其他div层。header层显示网页Logo，navhorisontell层显示页头部分的导航菜单，header层和navhorisontell层共同组合成页头部分。col1层作为页面主体的中间部分，显示段落信息；col2是页面主体的右侧部分，显示了一张图片；nav是页面主体中的左侧部分，显示的是导航链接。footer用于页脚部分，存放版权和地址信息。content层是层col1和层col2的父层，<link>标记表示当前HTML页面引入了一个CSS文件。

在IE9.0中，浏览效果如图15-6所示，可以看到页面中显示列表显示、段落信息和图片，其显示顺序依据HTML网页的DIV排列顺序。




图15-5 个人网站首页



图15-6 基本HTML页面显示

03 添加CSS代码，修饰全局样式。

```
*{margin:0px;padding:0px}/* 布局*/
```

```
body
{
background:url(Img/bodybg.jpg) repeat-x top center #eceddd;
font-family:Arial, Verdana,'Lucida Grande', Helvetica, sans-serif;
text-align: center;
color: #333333;
}

#wrapper
{
background-color:#fff;
margin-top: 20px;
margin-right: auto;
margin-bottom:0px;
margin-left: auto;
width:902px;
border:10px solid #ffffff;
}
```

上面代码定义了*全局选择器，body标记选择器和ID选择器wrapper。在body标记选择器中，定义了背景图片、字体、对齐方式和字体颜色等。ID选择器wrapper中定义了背景色、外边距、宽度和边框样式等。

在IE9.0中，浏览效果如图15-7所示，可以看到列表、段落信息和图片等都居中显示，网页上、左、右三侧都有边框显示。

04 添加CSS代码，修饰页头部分。

```
#header
{
background:url(Img/bgheader.jpg) no-repeat;
width:902px;
height:203px;
padding-top: 0px;
margin-left:0px;
margin-right:0px;
margin-top: 0px;
margin-bottom: 0px;
}
#header h1
{
float:left;
font-size:2.9em;
padding-top:60px;
padding-left:37px;
font-family:Arial,verdana, sans-serif;
color:#37210c;
font-weight:bolder;
letter-spacing:-1px;
}
.orange
{color:#e67e1f;}
```

```
#navhorisontell
{
float:left;
list-style:none;
margin-bottom:0px;
margin-top:0px;
margin-left:0px;
width:902px;
background-color: #37210c;
}
#navhorisontell ul
{
list-style:none;
margin-bottom:0px;
margin-top:0px;
margin-left:0px;
}
#navhorisontell li
{
text-align:left;
float:left;
padding-left:0px;
padding-top:0px;
padding-bottom:0px;
}
#navhorisontell ul li a
{
display:block;
background-color:#37210c;
border-right:1px solid #fff;
line-height:2.5em;
margin-right:0px;
padding:8px 14px 8px 14px;
color: #ecf9ff;
font-weight:normal;
font-size: 0.8em;
text-decoration: none;
}
#navhorisontell  li a:hover
{
color: #ecf9ff;
background-color:#543e29;
}
#navhorisontell ul li .selected
{
color: #ecf9ff;
background-color:#e67e1f;
}
```

上面代码主要修饰网页的页头部分，其中ID选择器header定义页头背景图片，宽度、高度和外边距等。ID选择器navhorisontell定义层宽度、列表显示样式、层浮动显示和外边距距

离。其他选择器都是在这两个选择器的基础上，对层中的每个具体标记进行修饰，例如定义h1样式。

在IE9.0中，浏览效果如图15-8所示，可以看到页面顶部显示了一张背景图片，并显示了导航菜单，当鼠标放到菜单上，背景色会显示为灰色样式。

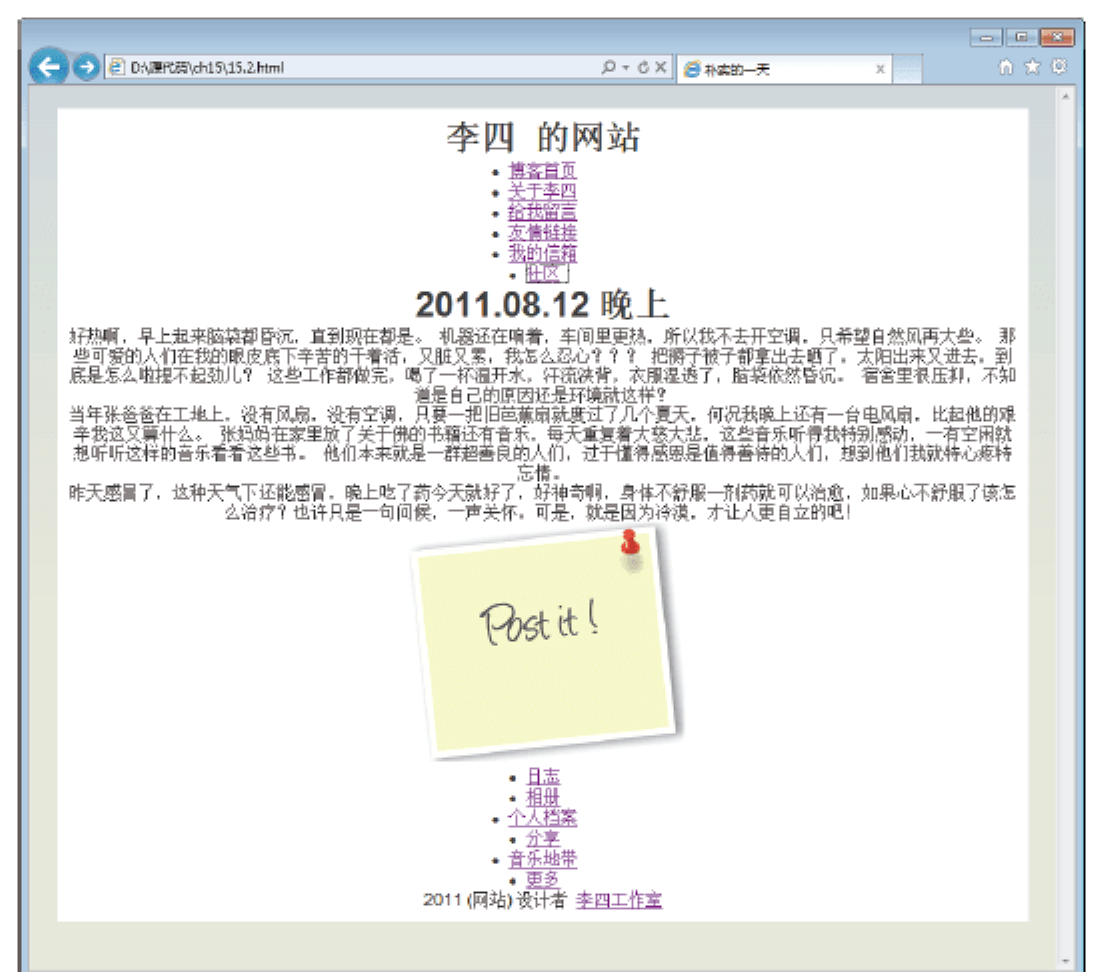


图15-7 定义全局样式



图15-8 定义页头部分

05 添加CSS代码，修饰页面主体左侧部分。

```
#nav
{
float:left;
list-style:none;
margin-top:15px;
margin-left:0px;
height:50%;
}
#nav ul
{
list-style:none;
margin-bottom:20px;
margin-top:20px;
margin-left:0px;
}
#nav li
{
text-align:left;
padding-left:0px;
padding-top:0px;
padding-bottom:0px;
border-bottom:1px solid #eaeada;
}
#nav ul li a
{
background-image: url(Img/bullet.gif);
```

```
background-repeat:no-repeat;
background-position:left center;
display:block;
background-color:#ffffff;
line-height:1.7em;
margin-right:0px;
padding-top:6px;
padding-bottom:6px;
padding-left:22px;
color: #666666;
font-weight:normal;
font-size: 0.8em;
text-decoration: none;
width:165px;
}
#nav  li a:hover
{
color: #37210c;
background-color:#f7f7f2;

}
#nav .selected
{
color: #37210c;
background-color:#f7f7f2;
}
```

在上面代码中，ID选择器nav定义页面主体左侧部分，例如定义层浮动布局，列表显示无特殊符号，上侧和左侧外边距距离，其层高占父div层50%。下面依据选择器nav，分别定义了无序列表显示样式，列表选项样式、超级链接显示样式、鼠标悬浮显示样式等。

在IE9.0中，浏览效果如图15-9所示，可以看到页面底部显示了一个导航菜单，其菜单前面都带有箭头标识。

06 添加CSS代码，修饰页面主体中间部分。

```
#content
{
float:right;
background-color:#ffffff;
width:710px;
margin-top:20px;
margin:auto;
padding:0px;
margin-bottom:30px;
margin-right:0px;
}
#col1
{
float:left;
width:410px;
margin-right:0px;
margin-top:20px;
```

```
    background-color:inherit;
    text-align:left;
    font-size:0.9em;
    padding:5px;
    }
    #col1 h1
    {
    display:block;
    font-size:0.9em;
    width:50px;
    font-family: arial;
    text-align:left;
    font-weight:bold;
    color:#403f3b;
    font-family:arial;
    font-weight:bold;
    padding:5px;
    margin-top:5px;
    margin-left:12px;
    }
    #col1 p
     {
    font: normal 0.9em Arial, Verdana, Helvetica, sans-serif;
    font-size:0.9em;
    color: #000000;
    padding:10px;
    text-align:left;
    }
```

在上面代码中，ID选择器content中定义了层在右边浮动显示，层宽度为710px以及背景色，外边距距离等。在ID选择器col1中定义了层在左边浮动显示，宽度为410px，字体大小、对齐方式依据内外边距等。下面选择器分别定义了col1层中的标题和段落信息。

在IE9.0中，浏览效果如图15-10所示，可以看到页面主体左侧显示了导航菜单，中间显示了段落信息，右侧显示的是一张图片信息。

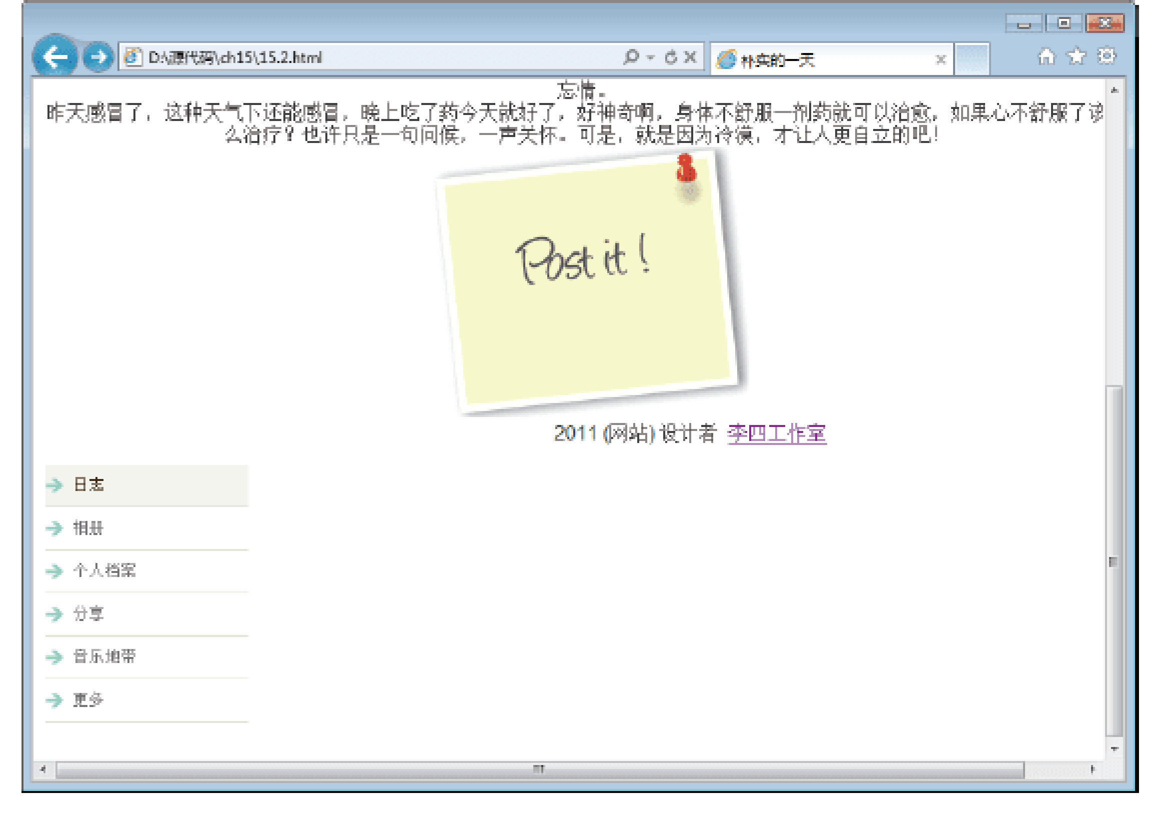



图15-9 定义页面主体左侧部分



图15-10 定义页面主体中间样式

07 添加CSS代码，修饰页面主体右侧部分。

```
#col2
{
float:right;
background-color:#ffffff;
width:272px;
margin-top:20px;
padding:8px 0 8px 8px;
text-align:left;
font-size:0.9em;
}
#col2 p
 {
font: bold 0.9em Arial, Verdana, Helvetica, sans-serif;
font-size:0.8em;
color: #000000;
padding:10px;
text-align:left;
}
#col2 .imagecol2
{
padding-left:0px;
padding-top:0px;
border:none;
}
```

上面代码中，ID选择器col2定义层在右边浮动显示，宽度为272px，字体大小、对齐方式、背景色和内外边距等。下面选择器定义了col2层中的段落和图片样式。由于本样式是对图片进行修饰，图片样式变化不大，这里就不再显示了。

08 添加CSS代码，修饰页脚部分。

```
#footer
{
width:902px;
height: 85px;
clear:both;
margin-top: 0px;
background-color:#dfeef9;
color:#666666;;
margin-left:auto;
margin-right:auto;
margin-bottom: 0px;
padding-top: 15px;
padding-right: 0px;
padding-bottom: 0px;
padding-left: 0px;
}
#footer p
{
font-size:0.7em;
```

```
font-family:arial;
font-weight:normal;
line-height: 1.4em;
color:#555555;
padding:25px 0 0 10px;
text-align:center;
}
#footer a
{
font-size:1em;
text-decration:none;
font-weight:normal;
color:#467AA7;
text-align:center;
}
#footer a:hover
{
text-decoration:underline;
font-weight:normal;
color:#467AA7;
text-align:center;
}
```

上面代码中，ID选择器footer中，定义了层宽度、高度、背景色、字体颜色和内外边距，并使用clear属性去掉float属性对本层的影响。下面选择器对footer层中的段落和超级链接样式进行了定义。

在IE9.0中，浏览效果如图15-5所示，可以看到页面底部显示了一个背景色为青蓝色的矩形，中间位置显示了版权信息。



15.4 综合实例3——旅游网站首页

人们生活富裕之后，就纷纷走出家门，开始去外地观光旅游。这时各大旅游景点的网站，就应运而生，用于介绍和宣传旅游特色。本实例模拟一个旅游网站首页，具体步骤如下所示。

01 分析需求。

作为旅游景点介绍的网站，除了正常的页面导航之外，还需要将旅游特色介绍出来，例如篝火晚会、看日出这些。实例完成后，页面效果如图15-11所示。

02 创建HTML页面，使用DIV分块。

根据上面效果可以看出，页面划分垂直方向上分为上中下三层结构，网页主体分为左右版式。其区域都可以通过div层来布局，代码如下：

```
    <html >
    <head>
    <title>夏季 之花</title>
    <link href="style1.css" rel="stylesheet" type="text/css" />
    </head>
    <body>
    <div id=all>
    <div id=header>
     <h1 class="website-title">夏季 之花 </h1>
    </div>
    <div>
    <p class=menu>
    <a class="menu-link" href="index.html">收费 </a> <a class="menu-link"
href="index.html">风景</a> <a class="menu-link" href="index.html">人文</a> <a
class="menu-link" href="index.html">礼物</a> <a class="menu-link" href="index.
html">联系 我们 </a>
    </p>
    </div>
    <div id=content>
    <div id=left>
    <h1>使用条款</h1>
              <p>你可以任意使用这个模板，而没有版权的限制。</p>
              <h1>夏季</h1>
              <p>夏季是四季之一。在大多数地区夏季是气候炎热、有可能缺水的季节，此外，由于
夏季时全球大部分学生都放暑假，因此也是相当旺盛的旅游季节。
     &lt;...&gt; </p>
              <p><em>在中国夏季从立夏(5月5日至7日之间）开始，到立秋结束；西方人则普遍称
夏至至秋分为夏季。在南半球，一般12月、1月和2月被定为夏季。从气候学意义上讲：连续五天平均温度
超过22℃算作夏季，直到五天平均温度低于22℃算作秋季。</em></p>
    </div>
    <div id=right>
    <p class="content-right">
    <a class="menu-right" href="#">夏天的太阳</a>
                <a class="menu-right" href="#">彩虹</a>
                <a class="menu-right" href="#">水</a>
              <a class="menu-right" href="#">海边沙滩</a>
       <a class="menu-right" href="#">篝火</a>
    </p>
    </div>
    </div>
    <div id=footer>
    <span class="made-in_text">设计者 <a class="made-in" title="" href="">李四工
作室</a></span>
    </div>
    </div>
    </body>
    </html>
```

上面HTML代码中，创建了all层作为布局容器，用来存放其他的div层。header层放置了页面Logo，menu层作为页面导航菜单，header层和menu层共同组成页头部分。content层是页

面主体部分，包含了两个子层，left层和right层。left层表示页面主体左侧部分，right层表示页面主体右侧部分。footer层表示页脚部分。“<link href="style1.css" rel="stylesheet" type="text/css" />”语句表示将style1.css样式文件引入到当前HTML文件中。

在IE9.0中，浏览效果如图15-12所示，可以看到网页自上而下显示了标题、超级链接和文本信息等，其中普通文字颜色为黑色，超级链接为蓝色。



图15-11 旅游网站首页

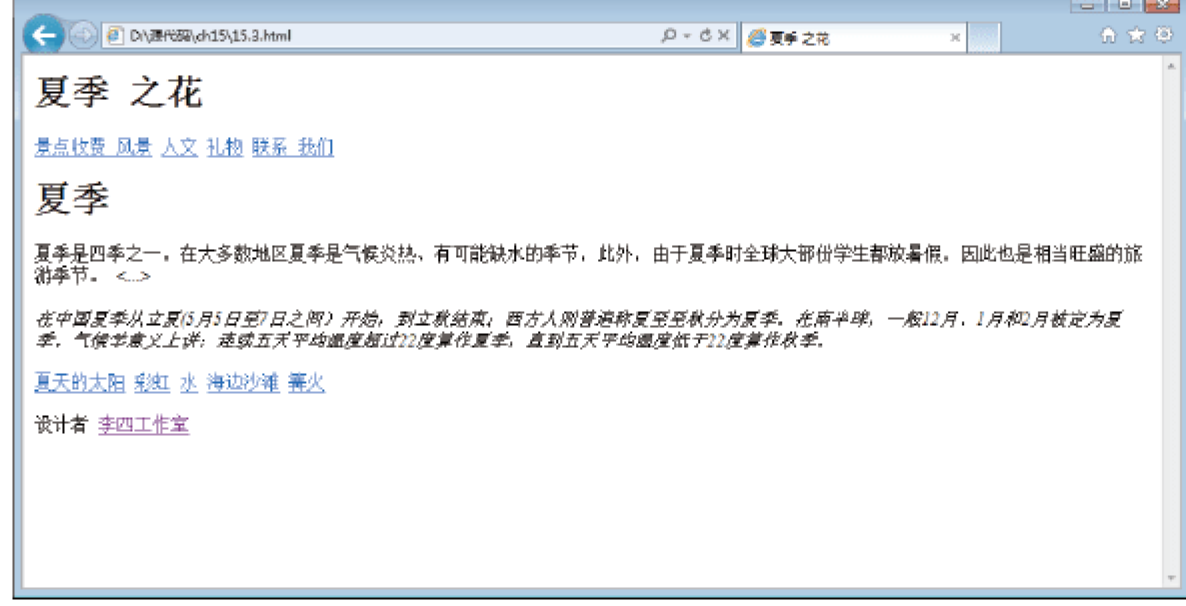


图15-12 基本HTML样式

03 添加CSS样式，定义全局样式。

```
*{
    padding:0px;
    margin:0px;
    }
body {
background-image:url(i/bg.gif);
margin:0px;
 padding-top:15px;
 font-family: sans-serif;
font-size:13px;
color:#222222;
text-align:center;
}
#all{
  width:960px;
   margin:0px auto 0px  auto;
   background:white;
}
```

上面代码中，body标记选择器定义了背景图片、内外边距以及文字大小、字体颜色和对齐方式等。ID选择器all中定义了层宽度为960px，背景色为白色，外边距均为0。

在IE9.0中，浏览效果如图15-13所示，可以看到页面加载了一个背景图片，并且HTML元素都在白色背景下居中显示。

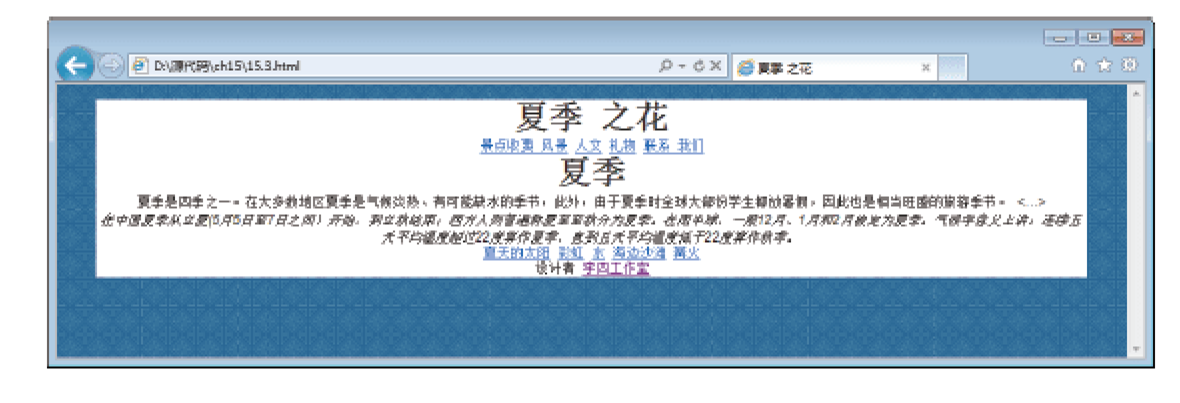


图15-13 定义全局样式

04 添加CSS样式，定义页头样式。

```
#header{
    width:960px;
    background-image: url(i/top-bg.jpg);
     height:250px;
}
.website-title {
position:relative;
left:0px;top:30px;
font-family: “幼圆”;
color: #FFFFFF;
text-align:center;
 font-size:34px;
letter-spacing:-1px;}
.menu {
height:35px;
background-image:url(i/menu-bg.gif);
position:relative;
width:960px;
text-align:left;
}
a.menu-link {
background-image:url(i/menu-sep.gif);
height:30px;
 line-height:30px;
background-position: right;
 background-repeat:no-repeat;
padding-left:7px;
padding-right:14px;
padding-top:6px;
padding-bottom:6px;
font-family: tahoma;
font-size:15px;
 color:#FFFFFF;
text-decoration:none;
 font-weight:bold}
a.menu-link:hover {
background-image:url(i/menu-bg_hover.gif);
background-position:center;
background-repeat:repeat-x;
color:#FFE7CF;}
```

在上面代码中，ID选择器header中，定义了层宽度、背景图片和高度。类选择器menu定义了背景图片、高度、宽度和对齐方式。类website-title定义了页头部分文字显示样式，例如字体大小、字符间距、颜色、字体、定位方式和左边位置。类menu-link选择器中定义超级链接的背景图片、内外边距、是否带有下划线等。

在IE9.0中，浏览效果如图15-14所示，可以看到页头部分，显示了背景图片和导航菜单。其中背景图片显示了文本信息“夏季之花”，导航菜单以黄色背景色显示。



图15-14 定义页头样式

05 添加CSS样式，定义页面主体样式。

```
#left{
        float:left;
        width:758px;
          text-align:left;
}
#left p{
     font-size:16px;
     line-height:30px;
     font-family:"宋体";
}
#right{
        float:right;
        width:198px;
}
a.menu-right, a.menu-right:visited {
font-size: 18px;
line-height:30px;
padding-left:20px;
text-decoration:none;
color:#aaaaaa;
background-color:#FAFAFA;
width:178px;
display:block;
 border-bottom:#CCCCCC
 solid 1px}
a.menu-right:hover {
background-color:#2679B7;
color:#D8E9F1}
```

上面代码中，left选择器定义了层浮动在左侧显示，宽度为758px，对齐方式为左对齐；right选择器定义了层浮动在右侧显示，宽度为198px。其他选择器分别定义了段落显示样式和超级链接显示样式。

在IE9.0中，浏览效果如图15-15所示，可以看到页面主体显示了两个区域：一个显示段落信息，一个显示列表信息。其中列表信息背景色为灰白色。

06 添加CSS样式，定义页脚样式。

```
#footer{
        clear:both;
         text-align:left;
}
a.made-in {color:#aaaaaa;}
.made-in_text {
    font-size: 11px;
    font-family: Verdana, Arial, Helvetica, sans-serif;
    color: #aaaaaa;
    padding-left:3px;
    padding-bottom:15px;

}
```

在上面代码中，footer选择器定义了对齐方式，并使用clear消除上面层使用float带来的影响。下面选择器定义了页脚部分超级链接显示样式。

在IE9.0中，浏览效果如图15-11所示，可以看到页脚部分以白色字体显示，并处在页面左下角位置。



图15-15 定义页面主体样式



15.5 专家解惑

1．在CSS3中，如果使用font-size:small语句设置字体，会发现字体大小在各个浏览器不同，这是为什么？

这是因为不同浏览器对字体大小small有不同的定义造成的，Firefox浏览器将small定义为13px，IE浏览器定义为16px，所以就导致了字体在不同浏览器中显示的大小不同。

其解决办法是在页面中设置字体大小的时候，不使用small设置大小，直接使用15px等具体的数值，这样，不管在何种浏览器中，都会以相同的大小显示，而不会出现差异。

2．在Firefox中设置鼠标样式，其属性值为hand，为什么不可以？

这是因为hand属性值仅在IE中可以使用，如果要设置兼容的手状，需要使用cursor: pointer语句，pointer属性值可以实现在IE和Firefox中显示游标手状。

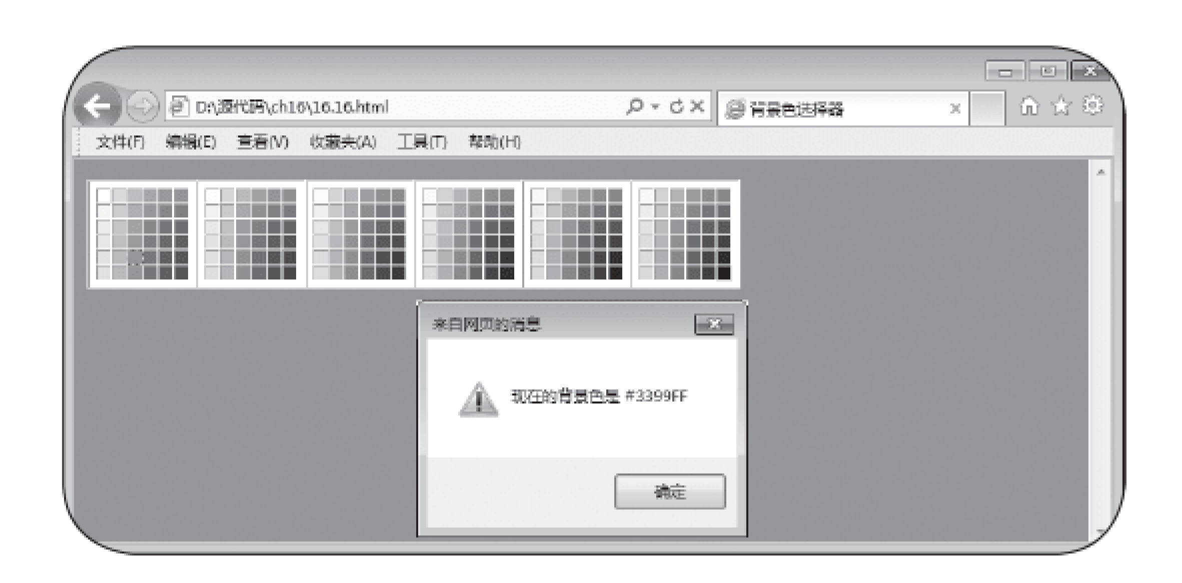


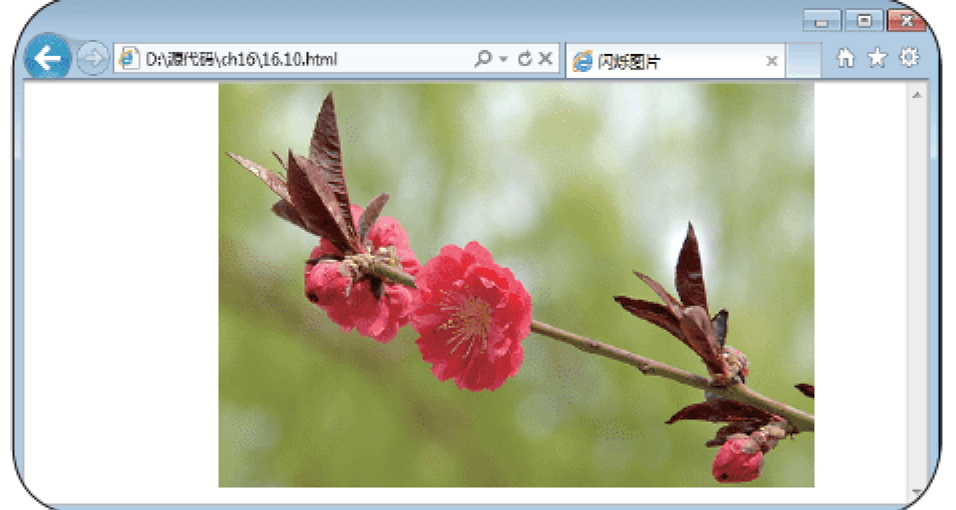
第 16 章

CSS3与JavaScript的搭配应用

本章引言

网页吸引人之处，莫过于具有动态效果，利用CSS伪类元素可以轻易实现超级链接的动态效果。不过利用CSS能实现的动态效果非常有限。在网页设计中，还可以将CSS与JavaScript结合创建出具有动态效果的页面。







16.1 JavaScript概述

JavaScript是一种面向对象、结构化、多用途的语言，它支持Web应用程序的客户端和服务器方面构件的开发。在客户端，利用JavaScript脚本语言，可以设计出多种网页特效，从而提高网页浏览量。

16.1.1 JavaScript概念

JavaScript是一种脚本编写语言，它采用小程序段的方式实现编程，像其他脚本语言一样，JavaScript同样也是一种解释性语言，它提供了一个简易的开发过程。JavaScript是一种基于对象的语言，同时也可以看作是一种面向对象的语言。这意味着它具有定义和使用对象的能力。因此，许多功能可以由脚本环境中对象的方法与脚本之间进行相互写作来实现。

JavaScript是动态的，它可以直接对用户或客户输入的数据做出及时响应，无须经过Web服务程序。它对用户请求的响应是以事件驱动的方式进行的。所谓事件驱动，就是指在主页中执行某种操作时所产生的动作，也称为“事件”。例如按下鼠标、移动窗口、选择菜单等都可以视为事件。当事件被触发时，可能会引起相应的事件响应处理。

JavaScript具有下面几个特点：

(1) 一种描述语言，通过浏览器就可以直接执行。

(2) 必须编写在HTML文件中，直接查看网页的源码，就可以看到JavaScript程序，所以没有保护，任何人都可以通过HTML文件复制代码。

(3) 结构较为自由松散，例如程序中使用变量前，并不需要明确定义，但对大小写敏感。

(4) 不具有读写文件及网络控制等功能。

16.1.2 JavaScript引用

在HTML文档中使用JavaScript有两种方式，一种是使用<script>标记将语句嵌入文档，另一种是将JavaScript做成独立的文件，引入到HTML文档中。

1. 嵌入JavaScript代码

如果在HTML文件中嵌入JavaScript代码，需要在<head>标记中间嵌入一个<script>标记，如下所示。

```
<script language="JavaScript">脚本程序</script>
```

此代码语句表示在HTML文档里嵌入了一段脚本程序，脚本语言为JavaScript。中间放置的是脚本程序。

【例16.1】实例文件：ch16\16.1.html

```
<html>
<head>
   <script language = "JavaScript">
      document.write("欢迎来到JavaScript动态世界");
  </script>
</head>
<body>
   <p>学习JavaScript！！！
</body>
</html>
```

该实例功能是在HTML文档里输出一个字符串，即“欢迎来到JavaScript动态世界”。

在IE9.0中，浏览效果如图16-1所示，可以看到网页输出了两句话，其中第一句就是JavaScript中的输出语句。

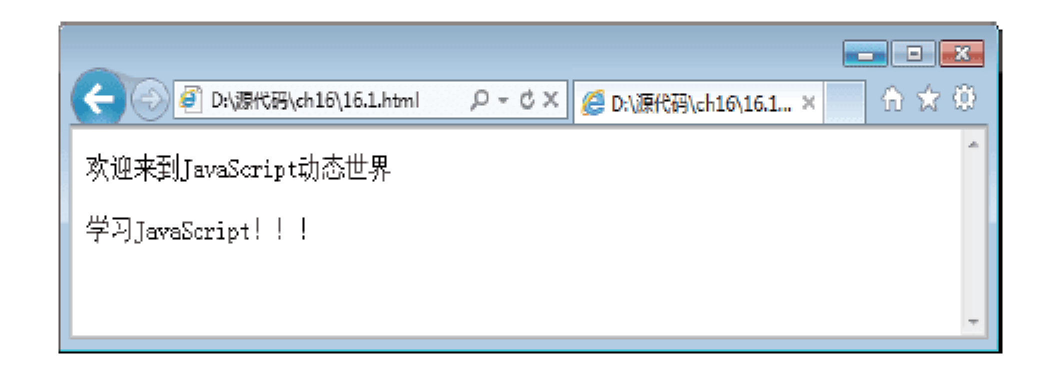


图16-1 嵌入JavaScript代码

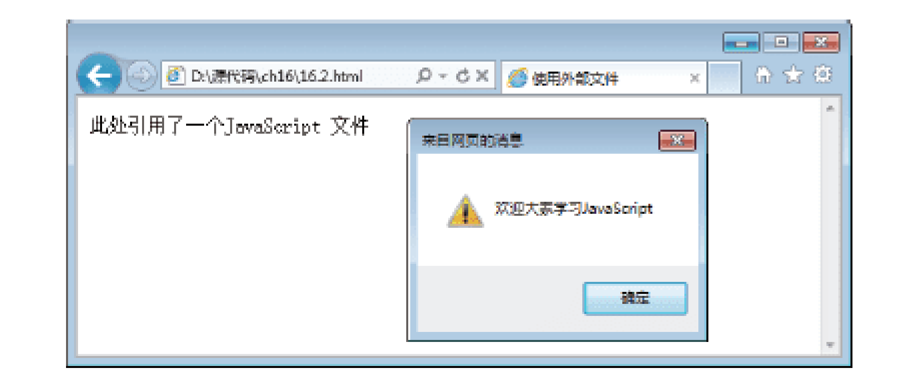


图16-2 导入JavaScript文件

在JavaScript的语法中，分号“;”是JavaScript程序作为一个语句结束的标识符。

2. 导入JavaScript文件

如果要导入JavaScript文件，需要将JavaScript做成一个独立文件，然后使用<script>标记将其引入到HTML文件中。其引入格式如下：

```
<script src = "hm_19.js"></script>
```

【例16.2】实例文件：ch16\16.2.html

```
<html>
<head>
<title>使用外部文件</title>
<script src = "hello.js"></script>
</head>
<body>
<P>此处引用了一个JavaScript 文件
</body>
</html>
```

在IE9.0中，浏览效果如图16-2所示，可以看到网页首先弹出一个对话框，显示提示信息。单击【确定】按钮后，会显示网页内容。

16.1.3 JavaScript与CSS

JavaScript和CSS都有一个共同的特点，就是二者都是在浏览器上解析并运行的。CSS用

于设置网页上的样式和布局，从而增加网页特效；而JavaScript是一种脚本语言，可以直接在网页上被浏览器解释运行。如果将JavaScript的程序和CSS的静态效果结合起来，可以创建出大量的动态特效，即使用JavaScript的循环、分支语句来控制CSS样式的不断转换。

对于动态特效，实际上在前面CSS已经有了实现，例如当鼠标停留在表格中某一行时，会显示一种颜色，不停留时会显示另外一种颜色。同样，对于其他的HTML元素，可以用JavaScript动态判断，从而调用不同的CSS样式。

JavaScript和CSS的结合运用，是喜爱网页特效浏览者的一大喜讯。作为一个网页设计者，通过对JavaScript和CSS的学习，可以创作出大量的网页特效。



16.2 JavaScript语法基础

JavaScript是一种脚本语言，并具有自己的数据类型、变量标识符、运算符和流程控制。但它是一种弱类型语言，非常容易学习和掌握。

16.2.1 数据类型

在JavaScript中有4种基本的数据类型：数值（整数和实数）、字符串型（用“”号或‘’括起来的字符或数值）、布尔型（使用true或false表示）和空值。此外，JavaScript还定义了其他复合数据类型，例如Date对象是一个日期和时间类型。

JavaScript具有的数据类型，如表16-1所示。

表16-1　JavaScript的基本数据类型

数据类型	数据类型名称	示例
number	数值类型	123、071（十进制）、0X1fa（十六进制）
string	字符串类型	'Hello'、'get the &'、'b@911.com'、"Hello"
object	对象类型	Date、Window、Document、Function
boolean	布尔类型	true和false
null	空类型	null
undefined	未定义类型	没有被赋值的变量所具有的“值”

1. 数值型

JavaScript的数值类型可以分为4类，即整数、浮点数、内部常量和特殊值。

整数可以为正数、0 或者负数；浮点数可以包含小数点、也可以包含一个“e”（大小写均可，在科学记数法中表示“10的幂”）、或者同时包含这两项。整数可以以10（十进制）、8（八进制）和16（十六进制）作为基数来表示。

内部常量和特殊值一般不常用，在这里就不再详细介绍。

2．字符串

字符串是由一对单引号（' '）或双引号（" "）和引号中的部分构成的。一个字符串也是JavaScript中的一个对象，有专门的属性。

引号中间的部分可以是任意多的字符，如果没有，则是一个空字符串。如果要在字符串中使用双引号，则应该将其包含在使用单引号的字符串中，使用单引号时则反之。

3．布尔型

布尔类型boolean表示一个逻辑数值，用于表示两种可能的情况。逻辑真，用true表示；逻辑假，用false来表示。通常，我们使用1表示真，0表示假。

4．undefined

未定义数据类型undefined表示变量被创建后，该变量未被赋过值，那么此时变量的值就是未定义数据类型。对于数字，未定义数值表示NaN；对于字符串，未定义数值表示undefined；对于逻辑数值，未定义数值表示为假。

5．null

在JavaScript里，使用null声明的变量并不是0。null是一个特殊的类型，它表示一个空值，即没有值，而不是0，0是有值的。

由于JavaScript采用弱类型的形式，因而一个数据的变量或常量不必首先做声明，而是在使用或赋值时确定其数据类型。当然也可以先声明该数据类型，它是通过在赋值时自动说明其数据类型的。

16.2.2 变量

在JavaScript中，使用var关键字来声明变量。语法格式如下：

```
var var_name;
```

JavaScript是一种区分大小写的语言，因此变量temp和变量Temp代表不同的含义。另外，在命名变量时必须遵循以下规则：

- 变量名由字母、数字、下划线和美元符组成。
- 变量名必须以字母、下划线（_）或美元符（$）开始。
- 变量名不能是保留字。

JavaScript语言使用等号（=）给变量赋值，等号左边是变量，等号右边是数值。对变量赋值的语法如下：

```
变量 =  值;
```

JavaScript里的变量分为全局变量和局部变量两种。其中局部变量就是在函数里定义的变量，在这个函数里定义的变量仅在该函数中有效。如果不写var，直接对变量进行赋值，那么JavaScript将自动把这个变量声明为全局变量。

使用示例如下：

```
var yourAppleNumber = 100
//等价于
var yourAppleNumber
yourAppleNumber = 100
```

16.2.3 运算符

在JavaScript的程序中要完成各种各样的运算，是离不开运算符的。它用于将一个或几个值进行运算而得出所需要的结果值。在JavaScript的常用的运算符有算术运算符、比较运算符、逻辑运算符。

1. 算术运算符

算术运算符是最简单、最常用的运算符，所以有时也称它们为简单运算符。可以使用它们进行通用的数学计算，如表16-2所示。

表16-2 算术运算符

运算符	说明	示例
+	加法运算符，用于实现对两个数字进行求和	x+100、100+1000、+100
-	减法运算符或负值运算符	100-60、-100
*	乘法运算符	100*6
/	除法运算符	100/50
%	求模运算符，也就是算术中的求余	100%30
++	将变量值加1后再将结果赋值给该变量	x++用于在参与其他运算之前先将自己加1后，再用新的值参与其他运算 ++x用于先用原值与其他运算后，再将自己加1
--	将变量值减1后再将结果赋值给该变量	x--、--x与++的用法相同

2. 比较运算符

比较运算符用于对运算符的两个表达式进行比较，然后根据比较结果返回布尔类型的值。例如，比较两个值是否相同或比较两个数字值的大小等。在表16-3中列出了JavaScript支持的比较运算符。

表16-3 比较运算符

运算符	说明	示例
==	判断左右两边表达式是否相等，当左边表达式等于右边表达式时返回true，否则返回false	Number == 100 Number1 == Number2
!=	判断左边表达式是否不等于右边表达式，当左边表达式不等于右边表达时返回true，否则返回false	Number != 100 Number1 != Number2
>	判断左边表达式是否大于右边表达式，当左边表达式大于右边表达式时返回true，否则返回false	Number > 100 Number1 > Number2
>=	判断左边表达式是否大于等于右边表达式，当边表达式大于等于右边表达式时返回true，否则返回false	Number >= 100 Number1 >= Number2

（续表）

运算符	说明	示例
<	判断左边表达式是否小于右边表达式，当左边表达式小于右边表达式时返回true，否则返回false	Number < 100 Number1 < Number2
<=	判断左边表达式是否小于等于右边表达式，当左边表达式小于等于右边表达式时返回true，否则返回false	Number <= 100 Numer1 <= Number2

3. 逻辑运算符

逻辑运算符通常用于执行布尔运算，它们常和比较运算符一起使用来表示复杂运算，这些运算涉及的变量通常不止一个，而且常用于if、while和for语句中。表16-4列出了JavaScript支持的逻辑运算符。

表16-4 逻辑运算符

运算符	说明	示例
&&	逻辑与，若两边表达式的值都为true，则返回true；任意一个值为false，则返回false	100>60 &&100<200，返回true 100>50&&10>100，返回false
||	逻辑或，只有表达式的值都为false时，才返回false	100>60||10>100，返回true 100>600||50>60，返回false
!	逻辑非，若表达式的值为true，则返回false；否则返true	!(100>60)返回false !(100>600)返回true

除了上面介绍的常用运算符外，JavaScript还支持条件表达式运算符“?”，这个运算符是个三元运算符，它有三个部分：一个计算值的条件和两个根据条件返回的真假值。格式如下：

```
条件 ? 表示式1 : 表达式2
```

在使用条件运算符时，如果条件为真，则表达式的值使用表达式1的值；否则使用表达式2的值。示例如下：

```
( x > y ) ? 100*3 : 11
```

如果x的值大于y值，则表达式的值为300；否则x的值小于或等于y值时，表达式的值为11。

【例16.3】实例文件：ch16\16.3.html

```
<html>
<head>   <script language = "JavaScript">
       var a=3;
       var b=5;
       var c=b-a;
        document.write(c+"<br>");
       if(a>b)
              { document.write("a大于b<br>");}
       else
              { document.write("a小于b<br>");}
      document.write(a>b?"2":"3");
   </script>
</head><body></body>
```

```
</html>
```

上面代码创建了两个变量a和b，变量c的值是b和a的差。下面使用if语句判断a和b的大小，并输出结果。最后使用了一个三元运算符，如果a>b，则输出2；否则输出3。
表示在网页中换行，“+”是一个连接字符串。

在IE9.0中，浏览效果如图16-3所示，可以看到网页输出了JavaScript语句执行结果。

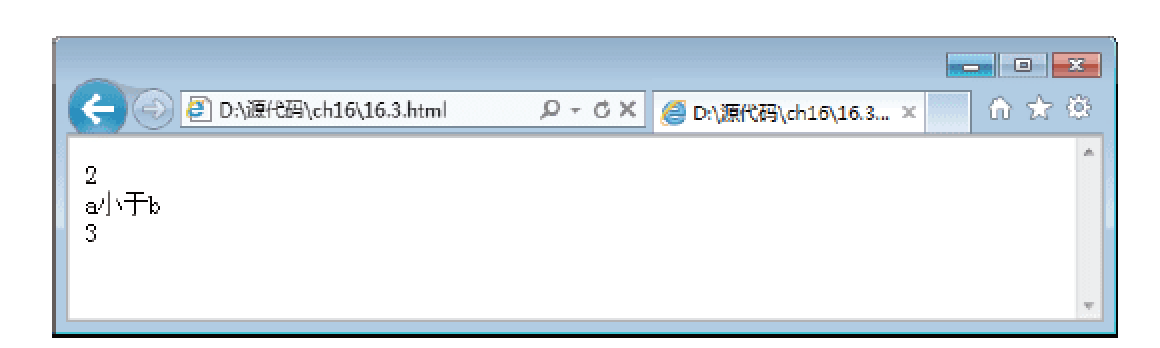


图16-3 运算符使用

16.2.4 流程控制语句

JavaScript编程中对流程的控制主要是通过条件判断、循环控制语句以及continue、break来完成的，其中条件判断按预先设定的条件执行程序，它包括if语句和switch语句；而循环控制语句则可以重复完成任务，它包括while语句、do while语句及for语句。

1. if语句

if语句是普遍使用的条件选择语句，每一种编程语言都有一种或多种形式的if语句，在编程中它是经常被用到的。

该语句格式如下：

```
if(条件语句)
{
    执行语句;
}
```

其中的“条件语句”可以是任何一种逻辑表达式，如果“条件语句”的返回结果为true，则程序先执行后面大括号对{}中的“执行语句”，然后接着执行它后面的其他语句；如果“条件语句”的返回结果为false，则程序跳过“条件语句”后面的“执行语句”，直接去执行程序后面的其他语句。大括号的作用就是将多条语句组合成一个复合语句，作为一个整体来处理，如果大括号中只有一条语句，这对大括号对{}可以省略。

2. if…else语句

if…else语句通常用于一个条件需要两个程序分支来执行的情况。

该语句格式如下：

```
if(条件语句)
{
    执行语句块1;
}
else
{
```

```
        执行语句块2;
    }
```

这种格式在if从句的后面添加一个else从句，这样当条件语句返回结果为false时，执行else后面部分的从句。

【例16.4】实例文件：ch16\16.4.html

```
    <html>
    <head>
       <script language = "JavaScript">
              var a="john";
              if(a!="john")
                    {
                       document.write("<h1 style='text-
align:center;color:red;'>欢迎JOHN光临</h1>");
                    }
              else{
                      document.write("<p style='font-size:15px;font-
weight:bolder;color:blue'>请重新输入名称</p>");
                    }
      </script>
    </head>
    <body>
    </body>
    </html>
```

上面代码中使用if…else语句，对变量a的值进行判断，如果a值不等于“john”则输出红色标题，否则输出蓝色信息。

在IE9.0中浏览效果如图16-4所示，可以看到网页输出了蓝色信息“请重新输入名称”。

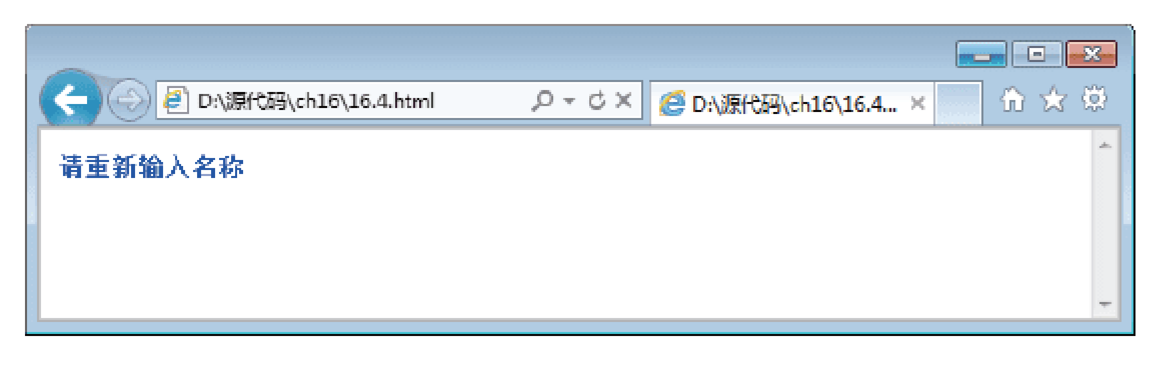


图16-4 if…else语句判断

3．switch选择语句

switch选择语句用于将一个表达式的结果同多个值进行比较，并根据比较结果选择执行语句。

该语句格式如下：

```
    switch (表达式)
    {
        case 取值1 :
           语句块1;break;
        case 取值2 :
           语句块2;break;
```

```
  ...
    case 取值n:
  语句块n;break;
    default :
      语句块n+1;
}
```

case语句只是相当于定义一个标记位置，程序根据switch条件表达式的结果，直接跳转到第一个匹配的标记位置处，开始顺序执行后面的所有程序代码，包括后面的其他case语句下的代码，直到碰到break语句或函数返回语句为止。default语句是可选的，它匹配上面所有case语句定义的值以外的其他值，也就是前面所有取值都不满足时，就执行default后面的语句块。

4．while语句

while语句是循环语句，也是条件判断语句。

该语句格式如下：

```
while(条件表达式语句)
{
  执行语句块
}
```

当“条件表达式语句”的返回值为true时，则执行大括号{}中的语句块，当执行完大括号{}中的语句块后，再次检测条件表达式的返回值，如果返回值还为true，则重复执行大括号{}中的语句块，直到返回值为false时，结束整个循环过程，接着往下执行while代码段后面的程序代码。

5．do…while语句

do…while语句的功能和while语句差不多，只不过它是在执行完第一次循环之后才检测条件表达式的值，这意味着包含在大括号中的代码块至少要执行一次，另外，do…while语句结尾处的while条件语句的括号后有一个分号“;”。

该语句格式如下：

```
do
{
  执行语句块
}while(条件表达式语句);
```

6．for语句

for语句通常由两部分组成，一部分是条件控制部分，一部分是循环部分。

该语句格式如下：

```
for(初始化表达式; 循环条件表达式; 循环后的操作表达式)
{
  执行语句块
}
```

在使用for循环前要先设定一个计数器变量，可以在for循环之前预先定义，也可以在使用时直接进行定义。在上述语法格式中，“初始化表达式”表示计数器变量的初始值；“循环条件表达式”是一个计数器变量的表达式，决定了计数器的最大值；“循环后的操作表达

式”表示循环的步长，也就是每循环一次，计数器变量值的变化，该变化可以是增大的，也可以是减小的，或进行其他运算。for循环是可以嵌套的，也就是在一个循环里还可以有另一个循环。

【例16.5】实例文件：ch16\16.5.html

```
    <html>
    <head>
       <script language = "JavaScript">
                 for(var i=0;i<5;i++){
                           document.write("<p style='font-size:"+i+"0px'>
欢迎学习JavaScript</p>");
                   }
      </script>
    </head>
    <body>
    </body>
    </html>
```

上面代码中，使用for循环输出了不同字体大小的语句。

在IE9.0中，浏览效果如图16-5所示，可以看到网页输出不同大小的语句，这些语句从小到大进行排列。

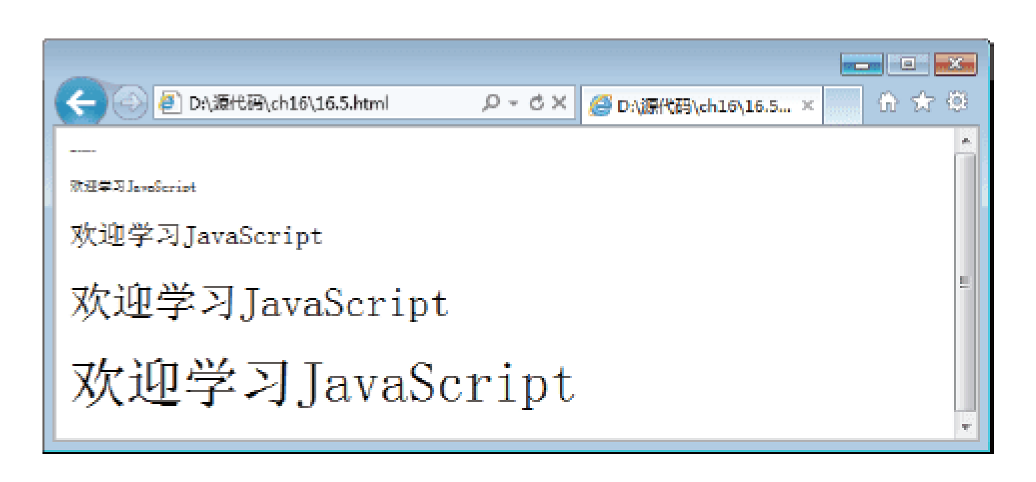


图16-5 for循环

除了上面的语句之外，JavaScript 还可以使用中断语句break和continue。break语句可以中止循环体中的执行语句和switch语句。一个无标号的break语句会把控制传给当前循环（while、do、for或switch）的下一条语句，如果有标号，控制会被传递给当前方法中的带有这一标号的循环语句。continue语句只能出现在循环语句（while、do、for）的循环体中，无标号的continue语句的作用是跳过当前循环的剩余部分，接着执行下一次循环。

16.2.5 函数

如果在一个程序中需要使用某个功能代码达到10次或更多，这时可以将这这个功能代码组成一个可以调用的函数，通过调用该函数来执行相应的语句，这样程序就将变得非常简洁，且便于后期进行维护。

在JavaScript中定义一个函数，必须以function关键字开头，函数名跟在关键字的后面，接着是函数参数列表和函数所执行的程序代码段。定义一个函数格式如下：

```
    function 函数名(参数列表)
```

```
{
    程序代码;
    return 表达式;
}
```

在上述格式中，“参数列表”表示在程序中调用某个函数时，一串传递到函数中的某种类型的值或变量，如果这样的参数多于一个，那么两个参数之间需要用逗号隔开。虽然有些函数并不需要接收任何参数，但在定义函数时也不能省略函数名后面的那对小括号，保留小括号中的内容为空即可。

另外，函数中的程序代码必须位于一对大括号之间，如果主程序要求返回一个结果集，就必须使用return语句，后面跟上这个要返回的结果。当然，return语句后可以跟上一个表达式，返回值将是表达式的运算结果。如果在函数程序代码中省略了return语句后的表达式，或者函数结束时没有return语句，这个函数就返回一个undefined的值。

【例16.6】实例文件：ch16\16.6.html

```
<html>
<head>
<title>计算器</title>
<script language="JavaScript" >
 function compute(op)
  {
     var num1=0;
     var num2=0;
     num1=parseFloat(document.myform.num1.value);
     num2=parseFloat(document.myform.num2.value);
     if (op=="+")
document.myform.result.value=num1+num2  ;
     if (op=="-")
document.myform.result.value=num1-num2  ;
     if (op=="*")
document.myform.result.value=num1*num2  ;
     if (op=="/"  &&  num2!=0)
document.myform.result.value=num1/num2  ;
  }
</script>
</head>
<body>
<form action="" method="post" name="myform" id="myform">
  <p>第一个数
    <input name="num1" type="text" id="num1" size="25">
    <br>
    第二个数
    <input name="num2" type="text" id="num2" size="25">
    </p>
  <p>
    <input name="addbutton" type="button" id="addbutton" value="  +  "
onclick="compute('+')">
    <input name="subbutton" type="button" id="subbutton" value="  -  "
onclick="compute('-')">
```

```
        <input name="mulbutton" type="button" id="mulbutton" value=" × "
onclick="compute('*')">
        <input name="divbutton" type="button" id="divbutton" value=" ÷ "
onclick="compute('/')">
      </p>
      <p>计算结果
        <input name="result" type="text" id="result" size="25">
      </p>
    </form>
    <p>  </p>
    </body>
    </html>
```

在IE9.0中，浏览效果如图16-6所示，可以看到网页输入两个不同的数值，可以求它们的和、差、积和商。

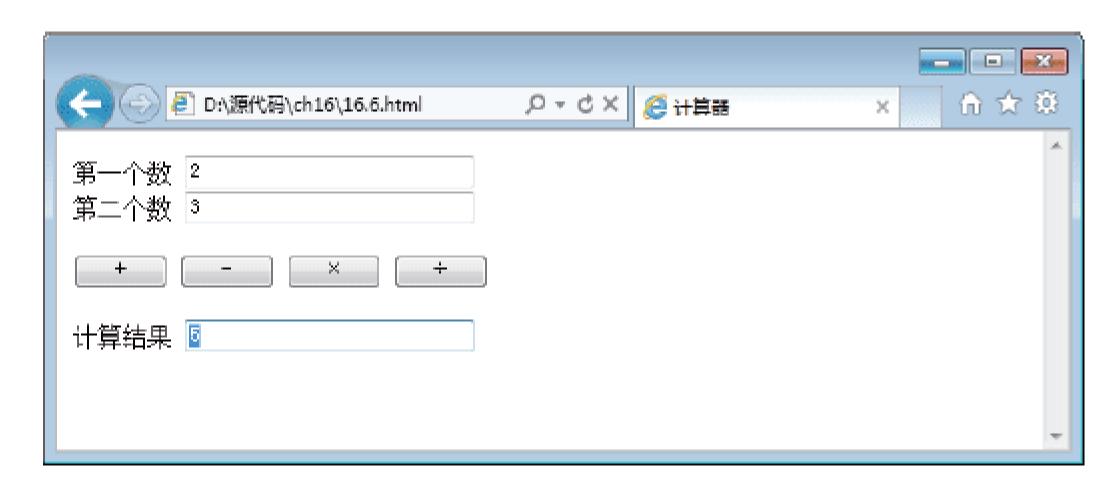


图16-6 使用函数

JavaScript不但允许用户根据自己的需要自定义函数，还支持大量的系统函数。常用的系统函数如表16-5所示。

表16-5 常用系统函数

函数名称	说明
eval()	返回字符串表达式中的值
parseInt()	返回不同进制的数，默认是十进制，用于将一个字符串按指定的进制转换成一个整数
parseFloat()	返回实数，用于将一个字符串转换成对应的小数
escape()	返回对一个字符串进行编码后的结果字符串
encodeURI	返回一个对URI字符串编码后的结果
decodeURI	将一个已编码的URI字符串解码成最原始的字符串返回
unescape ()	将一个用escape方法编码的结果字符串解码成原始字符串并返回
isNaN()	检测parseInt()和parseFloat()函数返回值是否为非数值型，如果是，返回true；否则，返回false
abs(x)	返回x的绝对值
acos(x)	返回x的反余弦值（余弦值等于x的角度），用弧度表示
asin(x)	返回x的反正弦值
atan(x)	返回x的反正切值
atan2(x,y)	返回复平面内点(x，y)对应的复数的幅角，用弧度表示，其值在-π～π之间
ceil(x)	返回大于等于x的最小整数

（续表）

函数名称	说明
cos(x)	返回x的余弦
exp(x)	返回e的x次幂（e^x）
floor(x)	返回小于等于x的最大整数
log(x)	返回x的自然对数（ln x）
max(a，b)	返回a、b中较大的数
min(a，b)	返回a、b中较小的数
pow(n，m)	返回n的m次幂（n^m）
random()	返回大于0小于1的一个随机数
round(x)	返回x四舍五入后的值
sin(x)	返回x的正弦值
sqrt(x)	返回x的平方根
tan(x)	返回x的正切值。isFinite()如果括号内的数字是“有限”的（介于 Number.MIN_VALUE和Number.MAX_VALUE之间）就返回true；否则返回false
toString()	用法：<对象>.toString()；把对象转换成字符串。如果在括号中指定一个数值，则转换过程中所有数值转换成特定进制



16.3 综合实例1——打字效果的文字

文字是网页的灵魂，没有文字的网页，不管特效多么绚丽多彩必定没有任何实际意义。文字特效始终是网页设计追求的目标，通过JavaScript可以实现多个网页特效。文字的打字效果是JavaScript脚本程序，将预先设置好的文字逐一在页面上显示出来。具体步骤如下所示。

01 分析需求。

如果要在网页中实现打字效果，需要创建一段预先设置好的文字，作为输出信息。本实例的效果如图16-7所示。

02 创建HTML页面，设置页面基本样式。

```
<html>
<head>
<title>打字效果的文字</title>
<style type="text/css">
body{font-size:14px;font-weight:bold;}
</style>
</head>
<body>
松风水月最新微博信息：<a id="HotNews" href="" target="_blank"></a>
</body>
</html>
```

上面代码中，在<head>标记中间，设置body页面的基本样式，例如字体大小为14px，字形加粗。并在body页面创建了一个超级链接。

在IE9.0中，浏览效果如图16-8所示，可以看到页面中只显示了一个提示信息。

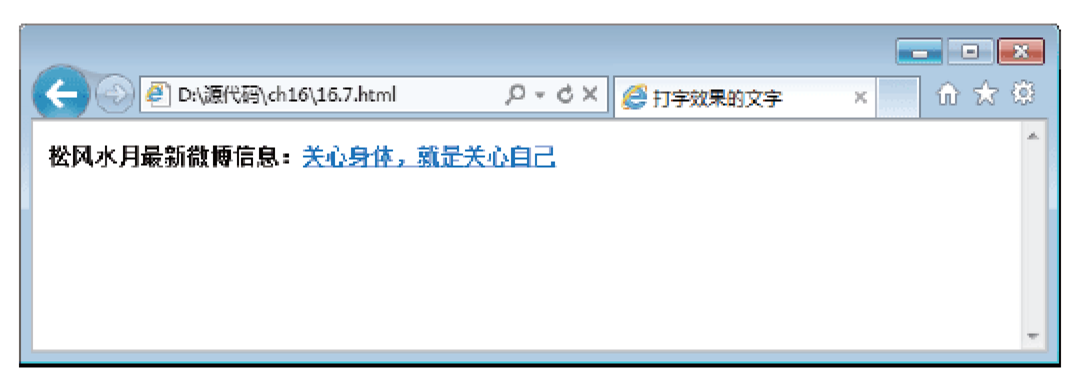


图16-7 打字效果

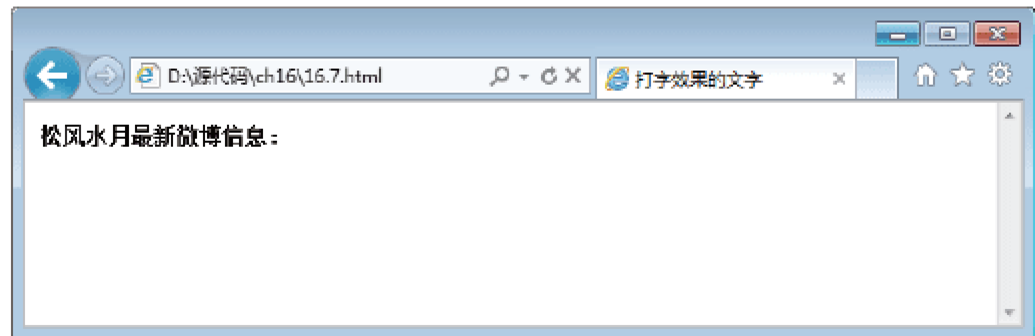


图16-8 页面基本样式

03 添加JavaScript代码，实现打字特效。

```
<script language="JavaScript">
<!--
var NewsTime = 2000;   //每条微博的停留时间
var TextTime = 50;     //微博文字出现等待时间，越小越快
var newsi = 0;
var txti = 0;
var txttimer;
var newstimer;
var newstitle = new Array();   //微博标题
var newshref = new Array();     //微博链接
newstitle[0] = "健康是身体的本钱";
newshref[0] = "#";
newstitle[1] = "关心身体，就是关心自己";
newshref[1] = "#";
newstitle[2] = "去西藏旅游了";
newshref[2] = "#";
newstitle[3] = "大雨倾盆，很大呀";
newshref[3] = "#";
function shownew()
{
  var endstr = "_"
  hwnewstr = newstitle[newsi];
  newslink = newshref[newsi];
  if(txti==(hwnewstr.length-1)){endstr="";}
  if(txti>=hwnewstr.length)
{
    clearInterval(txttimer);
    clearInterval(newstimer);
    newsi++;
    if(newsi>=newstitle.length)
{
      newsi = 0
    }
    newstimer = setInterval("shownew()",NewsTime);
    txti = 0;
    return;
```

```
        }
        clearInterval(txttimer);
        document.getElementById("HotNews").href=newslink;
        document.getElementById("HotNews").innerHTML = hwnewstr.
substring(0,txti+1)+endstr;
        txti++;
        txttimer = setInterval("shownew()",TextTime);
    }
    shownew();
    //-->
    </script>
```

因为上面的代码是一个整体，这里就不分开介绍了。上面的JavaScript代码中，主要调用shownew()函数完成打字效果。在JavaScript代码开始部分，定义了多个变量，其中数组对象newstitle用于存放文本标题。下面创建了shownew()函数，并在函数中通过变量和条件获取要显示的文字，通过“setInterval("shownew()",NewsTime)”语句输出文字内容。代码最后使用shownew()语句循环执行该函数中的输出信息。

在IE9.0中，浏览效果如图16-7所示，可以看到页面中，在提示信息后，每隔一定时间逐个打出单个文字，字体颜色为蓝色。



16.4 综合实例2——文字升降特效

有的网页为了加大广告宣传力度，往往在网页上设置一个自动升降的文字，用于吸引人的注意力。当单击这个升降文字时，会自动跳转到宣传页面。本实例将使用JavaScript和CSS实现文字升降效果。具体步骤如下所示。

01 分析需求。

如果需要实现文字升降，需要指定文字内容和文字升降范围，即为文字在HTML页面指定一个层，用于升降文字。实例完成后，实际效果如图16-9所示。

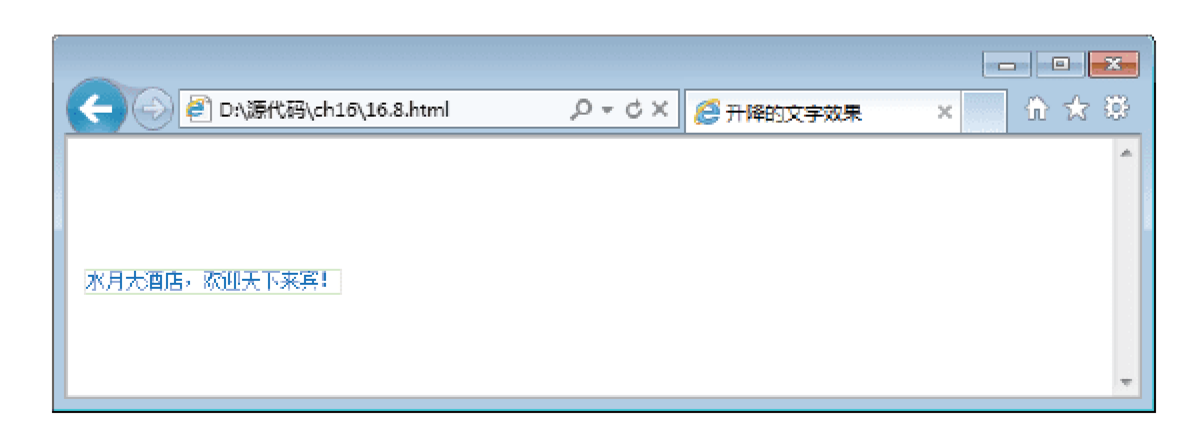


图16-9　文字升降

02 创建HTML，构建升降div层。

```
    <html>
    <head>
```

```
    <title>升降的文字效果</title>
    </head>
    <body>
    <div id="napis" style="position: absolute;top: -50;color: #000000;font-
family:宋体;font-size:9pt;border:1px #ddeecc solid">
    <a href="" style="font-size:12px;text-decoration:none;">
    水月大酒店，欢迎天下来宾！
    </a></div>
    <script language="JavaScript">
    <!--
    setTimeout('start()',20);
    //-->
    </script>
    </body>
    </html>
```

上面代码创建了一个div层，用于存放升降的文字，层的ID名称是napis，并在层的style属性中定义了层显示样式。例如字体大小、带有边框、字体等。在div层中，创建了一个超级链接，并设定了超级链接的样式。其中的JavaScript代码，用于定时调用start函数。

在IE9.0中，浏览效果如图16-10所示，可以看到页面空白，无文字显示。

03 添加JavaScript代码，实现文字升降。

```
    <script language="JavaScript">
    <!--
    done = 0;
    step = 4
    function anim(yp,yk)
    {
    if(document.layers) document.layers["napis"].top=yp;
    else document.all["napis"].style.top=yp;
    if(yp>yk) step = -4
    if(yp<60) step = 4
    setTimeout('anim('+(yp+step)+','+yk+')', 35);
    }
    function start()
    {
    if(done) return
    done = 1;
    if(navigator.appName=="Netscape")
    {
    var nap=document.getElementById("napis");
    nap.left=innerWidth/2 - 145;
    anim(60,innerHeight - 60)
    }
    else {
    napis.style.left=11;
    anim(60,document.body.offsetHeight - 60)
    }
    }//-->
    </script>
```

上面代码创建了函数anim()和start()，其中anim()函数用于设定每次升降的数值，start()函数用于设定每次开始的升降坐标。在IE9.0中，浏览效果如图16-9所示，可以看到页面中超级链接自动上下移动。

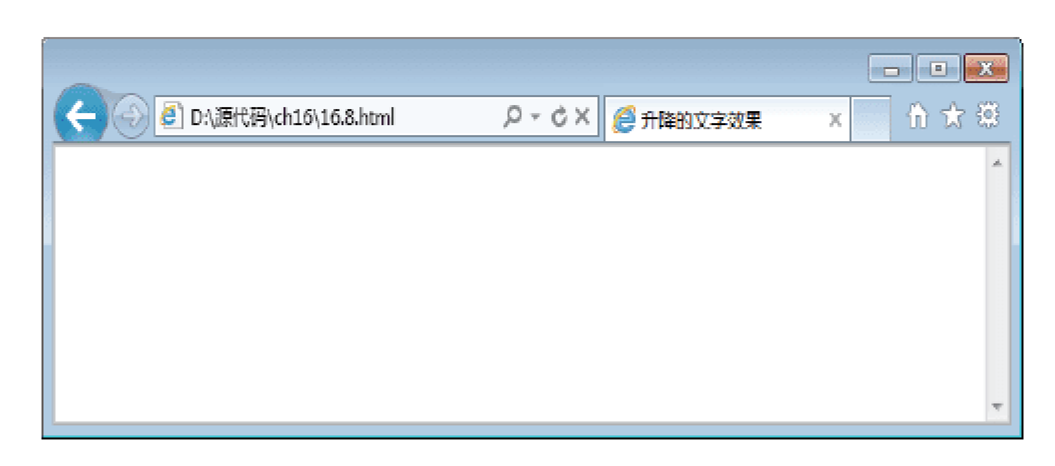


图16-10　空白页面



16.5　综合实例3——跑马灯效果

网页中有一种特效称为跑马灯，即文字从左到右自动输出，和晚上写字楼的广告霓虹灯非常相似。在网页中，如果CSS样式设计完美，就会设计出更加靓丽的网页效果，具体步骤如下所示。

01 分析需求。

完成跑马灯效果，需要使用JavaScript语言设置文字内容、移动速度和相应输入框，使用CSS设置显示文字样式。输入框用来显示水平移动文字。实例完成后，实际效果如图16-11所示。

02 创建HTML，实现输入表单。

```
<html>
<head>
<title>跑马灯</title>
   </head>
<body onLoad="LenScroll()">
<center>
<form name="nextForm">
<input type=text name="lenText">
</form>
</center>
</body>
```

上面代码非常简单，创建了一个表单，表单中存放了一个文本域，用于显示移动文字。

在IE9.0中，浏览效果如图16-12所示，可以看到页面中只是存在一个文本域，没有其他显示信息。

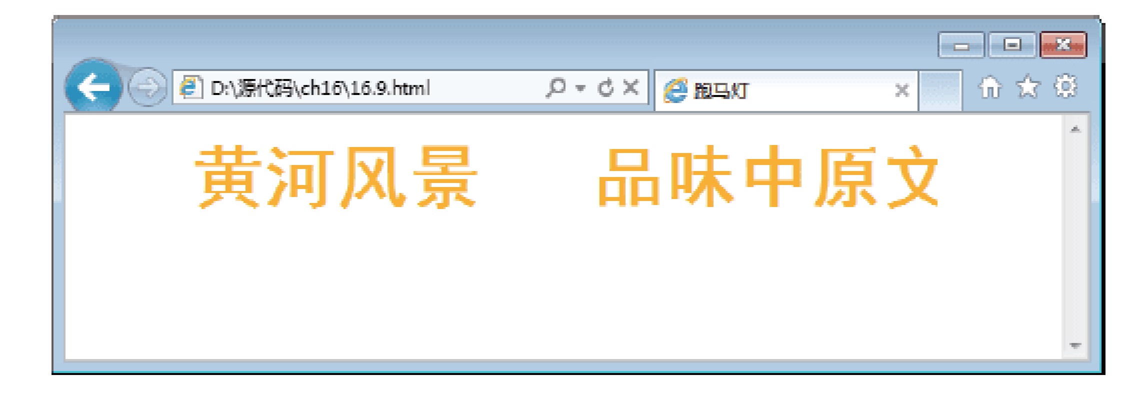


图16-11　马灯效果

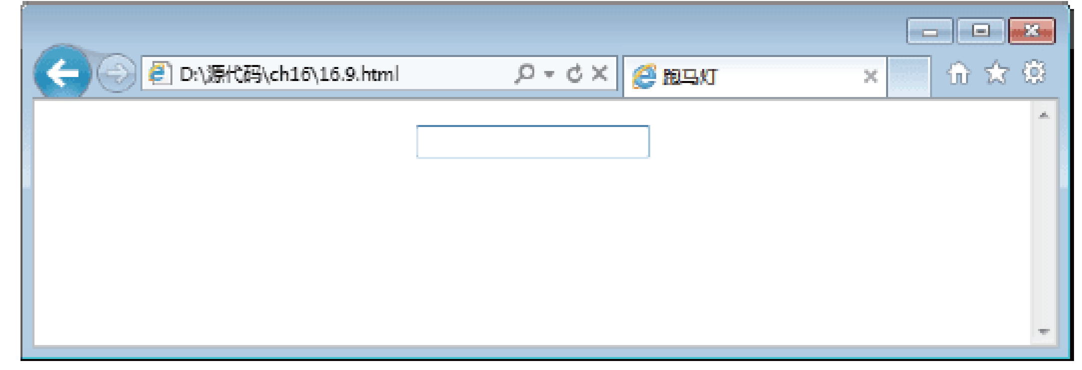


图16-12　实现基本表单

03 添加JavaScript代码，实现文字移动。

```
<script language="javascript">
var msg="品味中原文化，寄情黄河风景";          //移动文字
var interval = 400;                          //移动速度
var seq=0;

function LenScroll() {
  document.nextForm.lenText.value = msg.substring(seq, msg.length) + "   " + msg;
  seq++;
  if ( seq > msg.length )
    seq = 0;
  window.setTimeout("LenScroll();", interval);
}
</script>
```

上面代码中，创建了一个变量msg用于定义移动的文字内容，变量interval用于定义文字移动速度，LenScroll()函数用于在表单输入框中显示移动信息。

在IE9.0中，浏览效果如图16-13所示，可以看到输入框中显示了移动信息，并且从右向左移动。

04 添加CSS代码，修饰输入框和页面。

```
<style type="text/css">
<!--
body{
  background-color:#FFFFFF;   /* 页面背景色 */
}
input{
  background:transparent;     /* 输入框背景透明 */
  border:none;                /* 无边框 */
  color:#ffb400;
  font-size:45px;
  font-weight:bold;
  font-family:黑体;
}--></style>
```

上面代码设置了页面背景颜色为白色，在input标记选择器中，定义了边框背景为透明，即无边框。字体颜色为黄色，大小为45px，加粗并黑体显示。在IE9.0中，浏览效果如图16-11所示，可以看到，相比较原来页面字体变大，颜色为黄色，没有文本框显示。

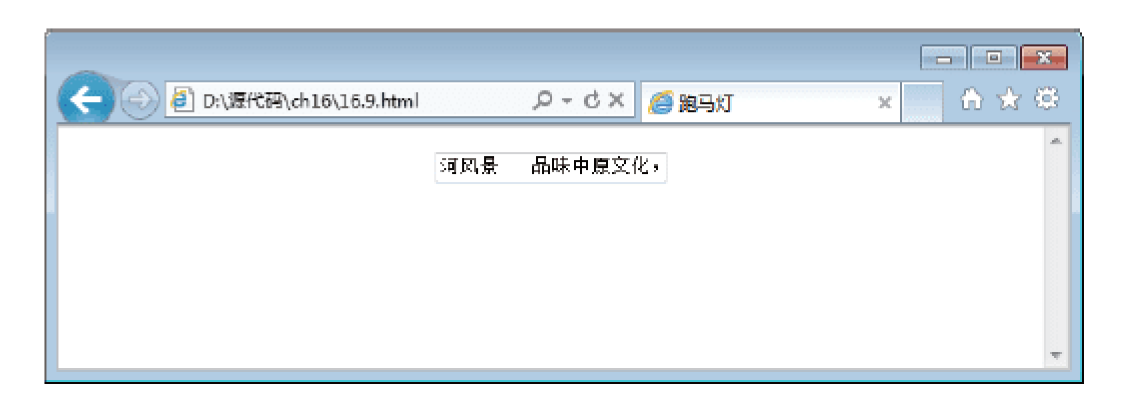


图16-13 实现移动效果



16.6 综合实例4——闪烁图片

图片闪烁是常用的一种特效，用JavaScript实现起来非常简单，只是需要注意时间间隔这个参数。数值越大闪烁越不连续，数值越小闪烁越厉害，可以随意更改这个值，直到取得满意的效果，具体步骤如下所示。

01 分析需求。

将图片放在一个DIV层上，设定图片为可见的，然后使用JavaScript程序代码设置DIV层的显示和隐藏，这样就达到了图片的闪烁效果。实例完成后，效果如图16-14所示。

02 创建HTML页面，构建DIV层。

```
<html>
<head>
<title>闪烁图片</title>
</head>
<body onload="socceronload()" topmargin="0">
<div id="soccer" style="position:absolute; left:150; top:0">
<a href="">
<img src="feng.jpg" border="0"></a>
</div>
</body>
</html>
```

上面代码中，创建了一个层，其ID名称为soccer，样式为绝对定位，坐标位置在（150,0）。然后在层中创建了一个图片，不带边框。

在IE9.0中浏览时，可以看到显示一张图片，不具有闪烁效果。



图16-14 图片闪烁

03 添加JavaScript代码，实现图片闪烁。

```
<script language="JavaScript">
var msecs = 500; //改变时间得到不同的闪烁间隔
var counter = 0;
function soccerOnload() {
setTimeout("blink()", msecs);
}
function blink() {
soccer.style.visibility =
(soccer.style.visibility == "hidden") ? "visible" : "hidden";
counter +=1;
setTimeout("blink()", msecs);
}
</script>
```

在JavaScript代码中，创建变量msecs用于定义闪烁时间间隔，创建变量counter用于计数。在函数soccerOnload()中，设定每隔指定时间图片闪烁一次，函数blink()用于设定图片显示，即图片是隐藏还是可见。

在IE9.0中，浏览效果如图16-14所示，可以看到显示一个图片，在指定时间内闪烁。



16.7 综合实例5——左右移动的图片

在广告栏中，经常会出现从右到左移动或者从左到右移动的、一张或者多张的图片。不但增加页面效果，也带来经济效益。本实例将使用JavaScript和CSS创建一张左右移动的图片。具体步骤如下所示。

01 分析需求。

实现左右移动的图片，需要再页面上定义一张图片，然后利用JavaScript程序代码获取图片对象，并使其在一定范围内，即水平方向上自由移动。实例的效果如图16-15所示。

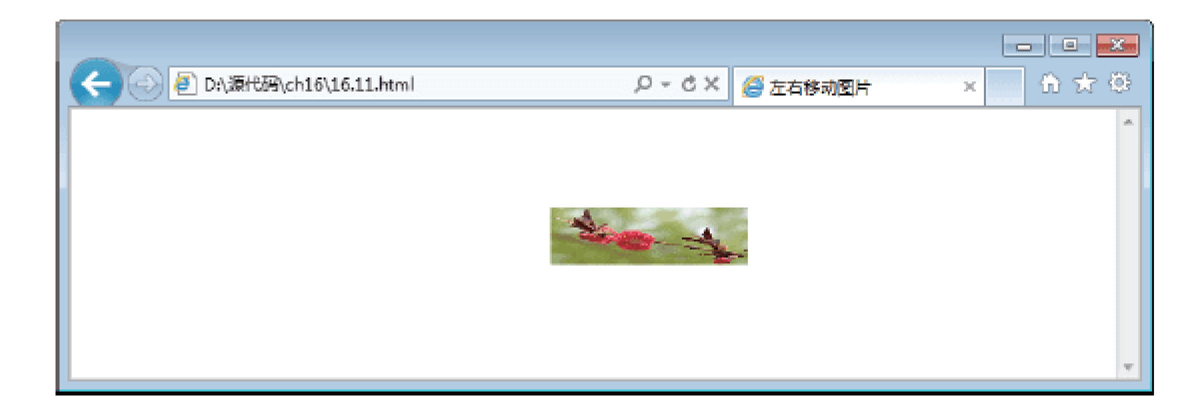


图16-15 图片移动

02 创建HTML页面，导入图片。

```
<html>
<head>
<title>左右移动图片</title>
</head>
<body>
<img src="feng.jpg" name="picture"
style="position: absolute; top: 70px; left: 30px;" BORDER="0" WIDTH="140"
HEIGHT="40">
<script language="JavaScript"><!--
setTimeout("moveLR('picture',300,1)",10);
//--></script>
</body>
</html>
```

上面代码中，定义了一张图片，图片是绝对定位，左边位置是（70,30）无边框，宽度为140px，高度为40px。JavaScript标记中，使用setTimeout方法，定时移动图片。

在IE9.0中，浏览效果如图16-16所示，可以看到网页上显示一张图片。

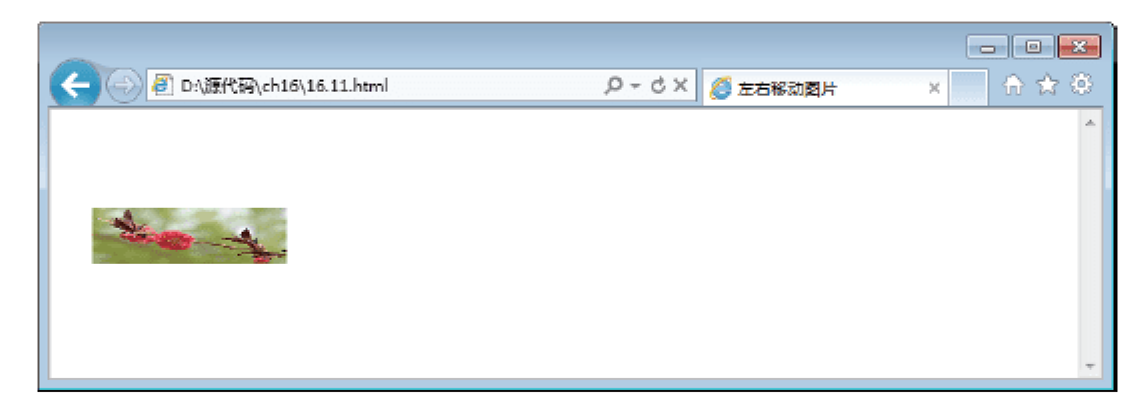


图16-16 图片显示

03 加入JavaScript代码，实现图片左右移动。

```
<script language="JavaScript"><!--
step = 0;
obj = new Image();
function anim(xp,xk,smer) //smer = direction
{
obj.style.left = x;
x += step*smer;
if (x>=(xk+xp)/2) {
if (smer == 1) step--;
else step++;
}
else {
```

```
if (smer == 1) step++;
else step--;
}
if (x >= xk) {
x = xk;
smer = -1;
}
if (x <= xp) {
x = xp;
smer = 1;
}
// if (smer > 2) smer = 3;
setTimeout('anim('+xp+','+xk+','+smer+')', 50);
}
function moveLR(objID,movingarea_width,c)
{
if (navigator.appName=="Netscape") window_width = window.innerWidth;
else window_width = document.body.offsetWidth;
obj = document.images[objID];
image_width = obj.width;
x1 = obj.style.left;
x = Number(x1.substring(0,x1.length-2)); // 30px -> 30
if (c == 0) {
if (movingarea_width == 0) {
right_margin = window_width - image_width;
anim(x,right_margin,1);
}
else {
right_margin = x + movingarea_width - image_width;
if (movingarea_width < x + image_width) window.alert("No space for
moving!");
else anim(x,right_margin,1);
}
}
else {
if (movingarea_width == 0) right_margin = window_width - image_width;
else {
x = Math.round((window_width-movingarea_width)/2);
right_margin = Math.round((window_width+movingarea_width)/2)-image_width;
}
anim(x,right_margin,1);
}
}
//--></script>
```

上面代码和文字水平方向移动的原理基本相同，只不过对象不同，这里就不再介绍了。

在IE9.0中，浏览效果如图16-15所示，可以看到网页上显示一张图片，并在水平方向上自由移动。



16.8 综合实例6——向上滚动的菜单

当网页包含信息比较多的时候，就需要设计一些导航菜单以方便页面导航。如果使用JavaScript 代码，将菜单做成动态效果，菜单将会更加吸引人。本实例将结合前面学习的内容，创建一个向上滚动的菜单，具体步骤如下所示。

01 分析需求。

实现菜单自动从下到上滚动，需要把握两个元素：一个是使用JavaScript实现要滚动的菜单，即导航栏；另一个是使用JavaScript控制菜单移动的方向。实例完成后效果如图16-17所示。

02 构建HTML页面。

```
<html>
<head>
<title>向上滚动的菜单</title>
</head>
<body bgcolor="#FFFFFF" text="#000000">
</body></html>
```

上面代码比较简单，只是实现了一个空白页面，页面的背景色为白色，前景色为黑色。在IE9.0中浏览效果如图16-18所示，可以看到显示了一个空白页面。

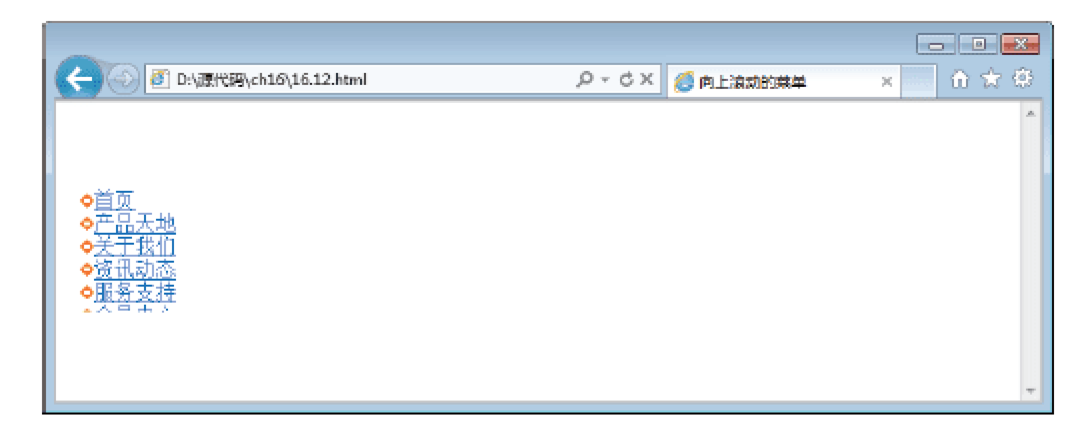


图16-17 菜单滚动

图16-18 空白HTML页面

03 加入JavaScript代码，实现菜单滚动。

```
<script language=javascript>
<!--
  var index = 9
  link = new Array(8);
  link[0] ='time1.htm'
  link[1] ='time2.htm'
  link[2] ='time3.htm'
  link[3] ='time1.htm'
  link[4] ='time2.htm'
  link[5] ='time3.htm'
  link[6] ='time1.htm'
  link[7] ='time2.htm'
```

```
      link[8] ='time3.htm'
      text = new Array(8);
      text[0] ='首页'
      text[1] ='产品天地'
      text[2] ='关于我们'
      text[3] ='资讯动态'
      text[4] ='服务支持'
      text[5] ='会员中心'
      text[6] ='网上商城'
      text[7] ='官方微博'
      text[8] ='企业文化'
      document.write ("<marquee scrollamount='1' scrolldelay='100' direction=
 'up' width='150' height='150'>");
      for (i=0;i<index;i++)
      {
        document.write (" <img src='dian3.gif' width='12' height='12'><a
href="+link[i]+" target='_blank'>");
        document.write (text[i] + "</A><br>");
      }
      document.write ("</marquee>")
    // --></script>
```

上面代码创建了两个数组对象link和text，用来存放菜单链接对象和菜单内容，在下面的JavaScript代码中，使用<marquee>定义页面在垂直方向上下移动。

在IE9.0中浏览效果如图16-17所示，可以看到页面左侧有一个菜单，自下向上自由移动。



16.9 综合实例7——跟随鼠标移动的图片

在众多网站中，特别是游戏网站或小型商业网站，都喜欢使用图片跟随鼠标的特效，一方面可以在鼠标指针旁边加上网站说明的相关信息或者欢迎信息，另一方面也吸引人的注意力，使人们更加关注此类网站。本实例实现图片跟随鼠标移动的特效，具体步骤如下所示。

01 分析需求。

需要通过JavaScript获取鼠标指针的位置，并且动态的调整图片的位置。图片通过position的绝对定位，可以很容易得到调整。CSS的绝对定位是JavaScript调整页面元素常用的方法。实例完成后，效果如图16-19所示。

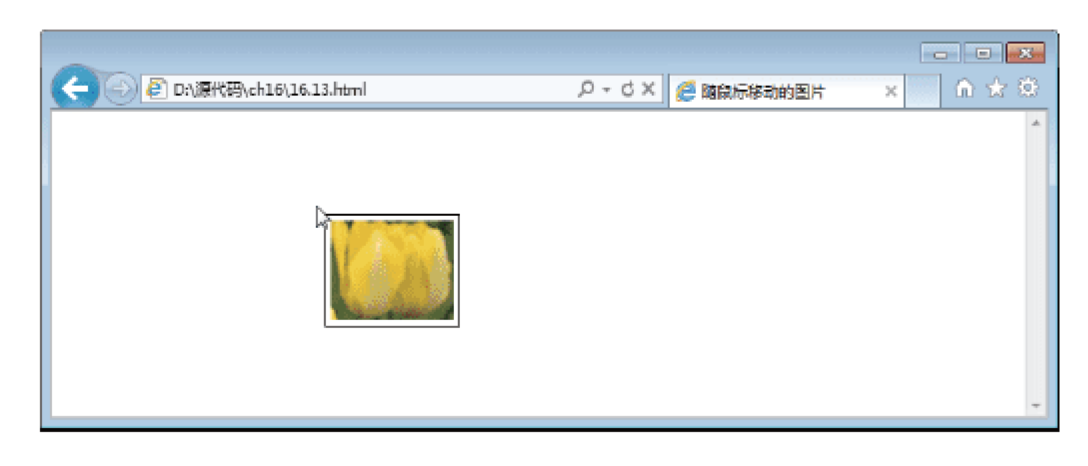


图16-19 图片跟随鼠标移动

02 创建基本HTML页面。

```
<html >
<head>
<title>随鼠标移动的图片</title>
</head>
<body>
</body>
</html>
```

上面代码比较简单，只是实现了一个HTML页面结构，这里就不再演示了。

03 添加JavaScript代码，实现图片跟随鼠标移动。

```
<script type="text/javascript">
function badAD(html)
{
    var ad=document.body.appendChild(document.createElement('div'));
    ad.style.cssText="border:1px solid #000;background:#FFF;position:abso
lute;padding:4px 4px 4px 4px;font: 12px/1.5 verdana;";
    ad.innerHTML=html||'This is bad idea!';
    var c=ad.appendChild(document.createElement('span'));
    c.innerHTML="×";
    c.style.cssText="position:absolute;right:4px;top:2px;cursor:pointer";
    c.onclick=function (){
        document.onmousemove=null;
        this.parentNode.style.left=’ -99999px’
    };
    document.onmousemove=function (e){
        e=e||window.event;
        var x=e.clientX,y=e.clientY;
        setTimeout(function() {
            if(ad.hover)return;
            ad.style.left=x+5+'px';
            ad.style.top=y+5+'px';
        },120)
}
    ad.onmouseover=function (){
        this.hover=true
    };
    ad.onmouseout=function (){
        this.hover=false
    }
}
badAD('<img src="18.png">')
</script>
```

上面代码中，使用appendChild()方法为当前页面创建了一个DIV对象，并为div层设置了相应样式。下面的e.clientX和e.clientY语句确定鼠标位置，并动态调整图片位置，从而实现图片移动效果。在IE9.0中，浏览效果如图16-19所示，可以看到鼠标在页面移动时，图片跟着移动。

16.10 综合实例8——树形菜单

作为一个首页，其特点之一是需要导航的页面很多，有时为了效果不得不将所有需要导航的部分都放到一个导航菜单中。树形导航菜单是网页设计中最常用的菜单之一。本实例将创建一个树形菜单，具体步骤如下所示。

01 分析需求。

实现一个树形菜单，需要三个方面配合，一个是<ul>无序列表，用于显示的菜单；一个是CSS样式，修饰树形菜单样式；一个是JavaScript 程序，实现单击时展开菜单选项。实例完成后，效果如图16-20所示。

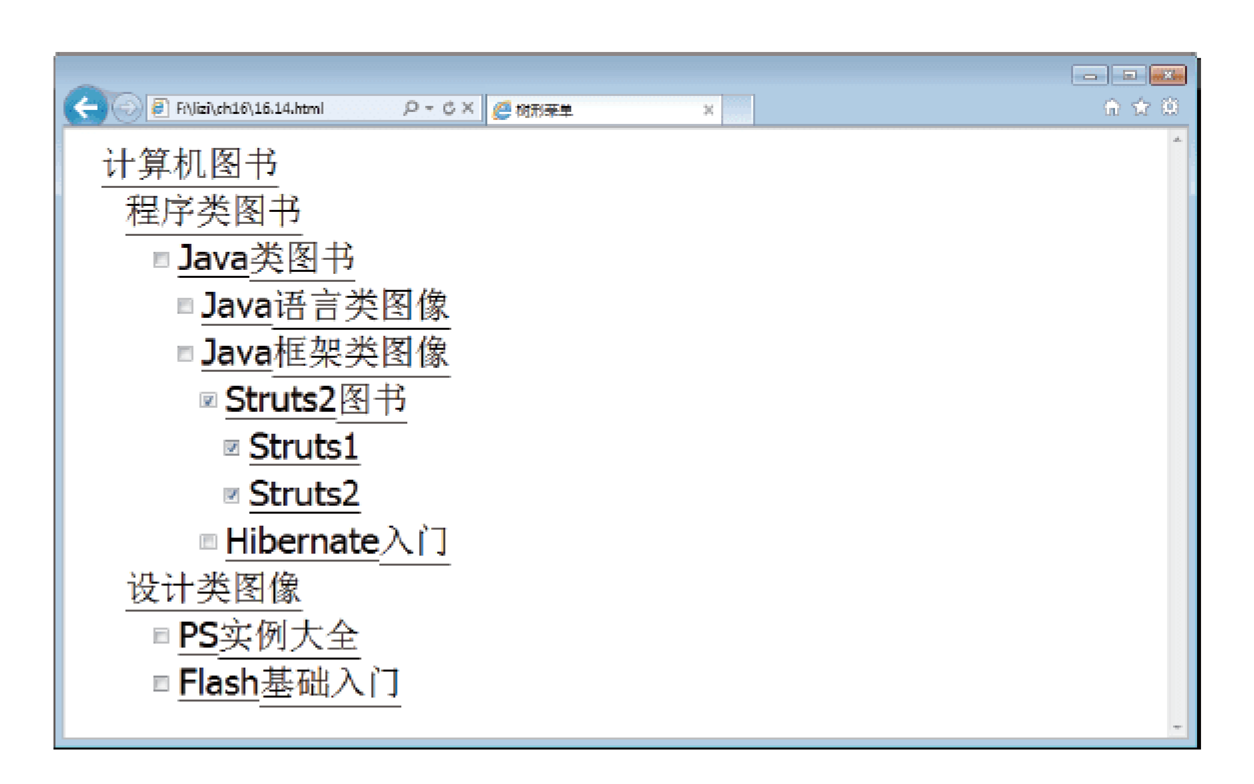


图16-20 树形菜单

02 创建HTML页面，实现菜单列表。

```
    <html >
    <head>
    <title>树形菜单</title>
    </head>
    <body>
    <ul id="menu_zzjs_net">
     <li>
      <label><a href="javascript:;">计算机图书</a></label>
      <ul class="two">
       <li>
        <label><a href="javascript:;">程序类图书</a></label>
        <ul class="two">
         <li>
          <label><input type="checkbox" value="123456"><a
href="javascript:;">Java类图书</a></label>
          <ul class="two">
           <li><label><input type="checkbox" value="123456"><a
href="javascript:;">Java语言类图像</a></label></li>
           <li>
```

```
            <label><input type="checkbox" value="123456"><a
href="javascript:;">Java框架类图像</a></label>
            <ul class="two">
             <li>
              <label><input type="checkbox" value="123456"><a
href="javascript:;">Struts2图书</a></label>
              <ul class="two">
               <li><label><input type="checkbox" value="123456"><a
href="javascript:;">Struts1</a></label></li>
               <li><label><input type="checkbox" value="123456"><a
href="javascript:;">Struts2</a></label></li>
              </ul>
             </li>
             <li><label><input type="checkbox" value="123456"><a
href="javascript:;">Hibernate入门</a></label></li>
            </ul>
           </li>
          </ul>
         </li>
        </ul>
       </li>
       <li>
        <label><a href="javascript:;">设计类图像</a></label>
        <ul class="two">
         <li><label><input type="checkbox" value="123456"><a
href="javascript:;">PS实例大全</a></label></li>
         <li><label><input type="checkbox" value="123456"><a
href="javascript:;">Flash基础入门</a></label></li>
        </ul>
       </li>
      </ul>
     </li>
    </ul>
    </body>
    </html>
```

在IE9.0中，浏览效果如图16-21所示，可以看到无序列表在页面上显示，并且显示全部元素，字体颜色为蓝色。

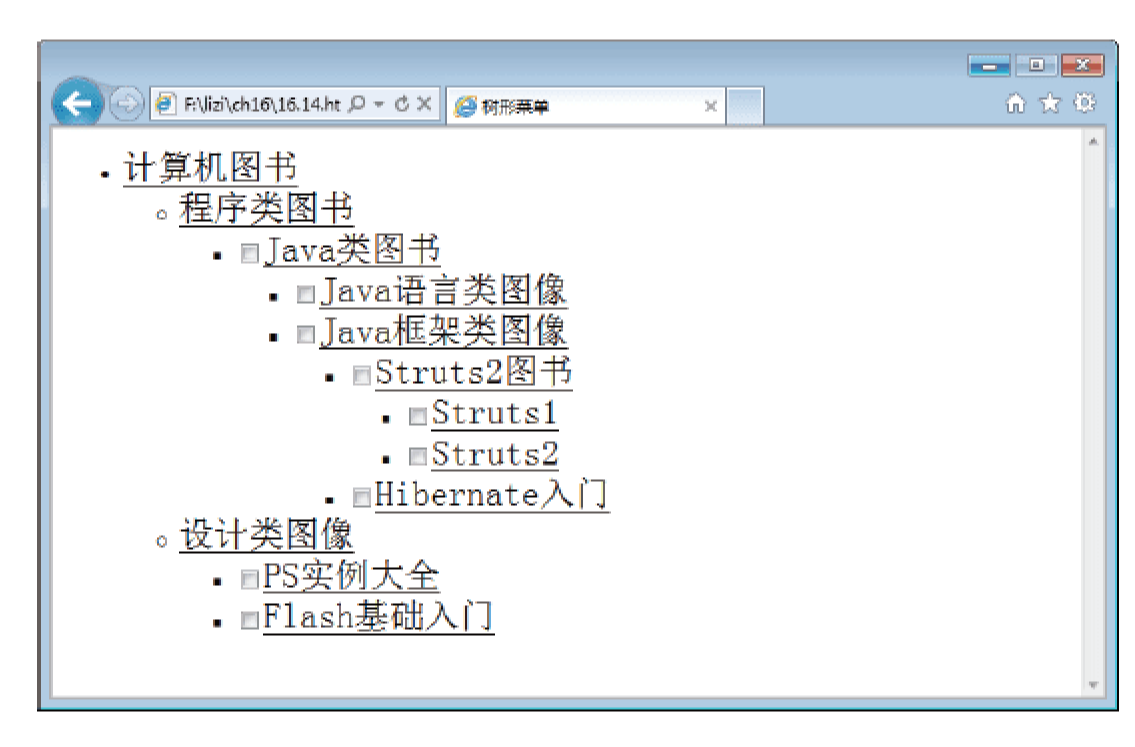


图16-21　无序列表

03 添加JavaScript代码，实现单击展开。

```
    <script type="text/javascript" >
     function addEvent(el,name,fn){//绑定事件
      if(el.addEventListener) return el.addEventListener(name,fn,false);
      return el.attachEvent('on'+name,fn);
     }
     function nextnode(node){//寻找下一个兄弟并剔除空的文本节点
      if(!node)return ;
      if(node.nodeType == 1)
       return node;
      if(node.nextSibling)
       return nextnode(node.nextSibling);
     }
     function prevnode(node){//寻找上一个兄弟并剔除空的文本节点
      if(!node)return ;
      if(node.nodeType == 1)
       return node;
      if(node.previousSibling)
       return prevnode(node.previousSibling);
     }
     function parcheck(self,checked){//递归寻找父亲元素，并找到input元素进行操作
      var par =  prevnode(self.parentNode.parentNode.parentNode.
previousSibling),parspar;
      if(par&&par.getElementsByTagName('input')[0]){
       par.getElementsByTagName('input')[0].checked = checked;
       parcheck(par.getElementsByTagName('input')[0],sibcheck(par.
getElementsByTagName('input')[0]));
      }
     }
     function sibcheck(self){//判断兄弟节点是否已经全部选中
      var sbi = self.parentNode.parentNode.parentNode.childNodes,n=0;
      for(var i=0;i<sbi.length;i++){
       if(sbi[i].nodeType != 1)//由于孩子结点中包括空的文本节点，所以这里累计长度的时候
也要算上去
        n++;
       else if(sbi[i].getElementsByTagName('input')[0].checked)
        n++;
      }
      return n==sbi.length?true:false;
     }
     addEvent(document.getElementById('menu_zzjs_net'),'click',function(e){
    //绑定input点击事件，使用menu_zzjs_net根元素代理
      e = e||window.event;
      var target = e.target||e.srcElement;
      var tp = nextnode(target.parentNode.nextSibling);
      switch(target.nodeName){
       case 'A'://点击A标签展开和收缩树形目录，并改变其样式选中checkbox
        if(tp&&tp.nodeName == 'UL'){
         if(tp.style.display != 'block' ){
          tp.style.display = 'block';
```

```
          prevnode(target.parentNode.previousSibling).className = 'ren'
         }else{
          tp.style.display = 'none';
          prevnode(target.parentNode.previousSibling).className = 'add'
         }
        }
        break;
       case 'SPAN'://点击图标只展开或者收缩
        var ap = nextnode(nextnode(target.nextSibling).nextSibling);
        if(ap.style.display != 'block' ){
         ap.style.display = 'block';
         target.className = 'ren'
        }else{
         ap.style.display = 'none';
         target.className = 'add'
        }
        break;
       case 'INPUT'://点击checkbox，父亲元素选中，则孩子节点中的checkbox也同时选中；
孩子结点取消，父元素随之取消
        if(target.checked){
         if(tp){
          var checkbox = tp.getElementsByTagName('input');
          for(var i=0;i<checkbox.length;i++)
           checkbox[i].checked = true;
         }
        }else{
         if(tp){
          var checkbox = tp.getElementsByTagName('input');
          for(var i=0;i<checkbox.length;i++)
           checkbox[i].checked = false;
         }
        }
        parcheck(target,sibcheck(target));//当孩子结点取消选中的时候，调用该方法递
归，其父节点的checkbox逐一取消选中
        break;
      }
     });
     window.onload = function(){//页面加载时给有孩子结点的元素动态添加图标
      var labels = document.getElementById('menu_zzjs_net').
getElementsByTagName('label');
      for(var i=0;i<labels.length;i++){
       var span = document.createElement('span');
       span.style.cssText ='display:inline-block;height:18px;vertical-align:
middle;width:16px;cursor:pointer;';
       span.innerHTML = ' '
       span.className = 'add';
       if(nextnode(labels[i].nextSibling)&&nextnode(labels[i].nextSibling).
nodeName == 'UL')
        labels[i].parentNode.insertBefore(span,labels[i]);
       else
        labels[i].className = 'rem'
```

```
        }
      }
    </script>
```

在IE9.0中，浏览效果如图16-22所示，可以看到无序列表在页面上显示，使用鼠标单击可以展开或关闭相应的选项，但其样式非常难看。

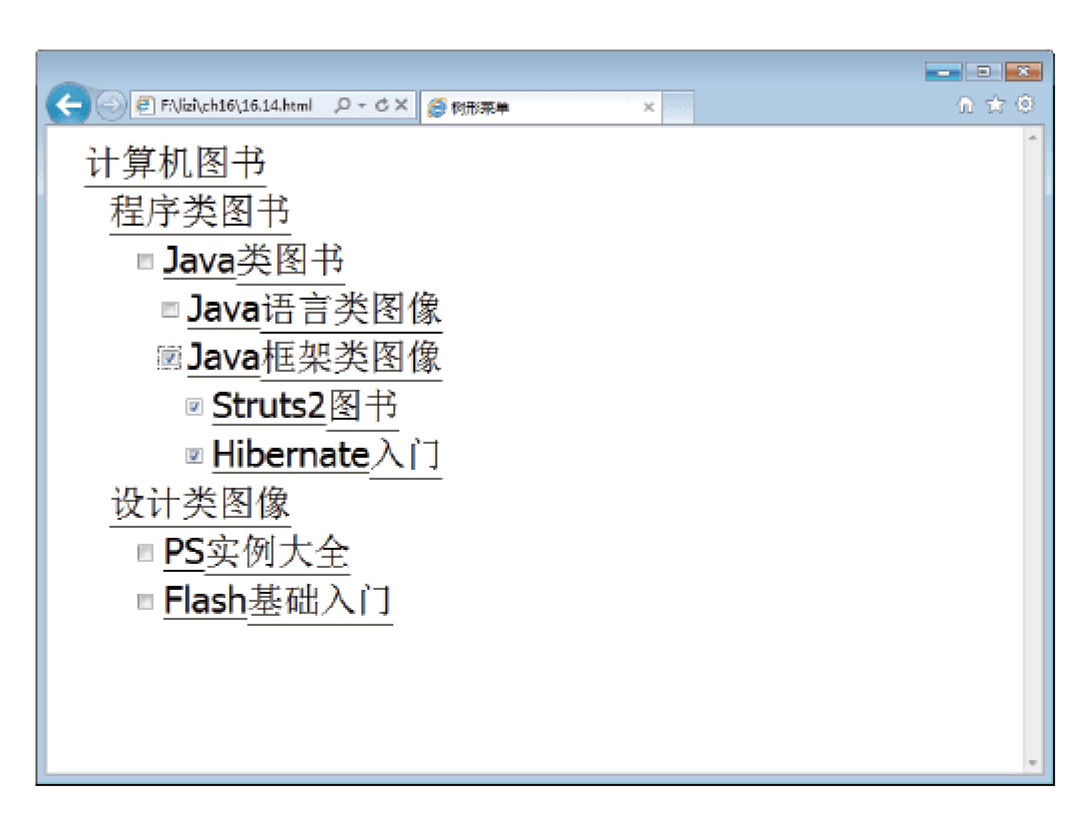


图16-22 实现鼠标单击事件

04 添加CSS代码，修饰列表选项。

```
    <style type="text/css">
    body{margin:0;padding:0;font:12px/1.5 Tahoma,Helvetica,Arial,sans-serif;}
    ul,li,{margin:0;padding:0;}
    ul{list-style:none;}
    #menu_zzjs_net{margin:10px;width:200px;overflow:hidden;}
    #menu_zzjs_net li{line-height:25px;}
    #menu_zzjs_net .rem{padding-left:16px;}
    #menu_zzjs_net .add{background:url() -4px -31px no-repeat;}
    #menu_zzjs_net .ren{background:url() -4px -7px no-repeat;}
    #menu_zzjs_net li a{color:#666666;padding-left:5px;outline:none;blr:expres
sion(this.onFocus=this.blur());}
    #menu_zzjs_net li input{vertical-align:middle;margin-left:5px;}
    #menu_zzjs_net .two{padding-left:20px;display:none;}
    </style>
```

在IE9.0中，浏览效果如图16-20所示，相比较原来的页面，可以看到样式变得非常漂亮。



16.11 综合实例9——时钟特效

在HTML5技术中，新增了一个容器画布canvas，用来在页面上绘制一些图形，利用这个新的特性，可以在网页中创建一个类似于钟表的特效。本实例结合第2章知识，创建了一个

时钟特效。具体步骤如下所示。

01 分析需求。

在画布上绘制时钟，需要绘制几个必要的图形，即表盘、时针、分针、秒针和中心圆。将上面几个图形组合起来，构成一个时针界面，然后使用JavaScript代码，根据时间移动秒针、分针和时针。实例完成后，效果如图16-23所示。

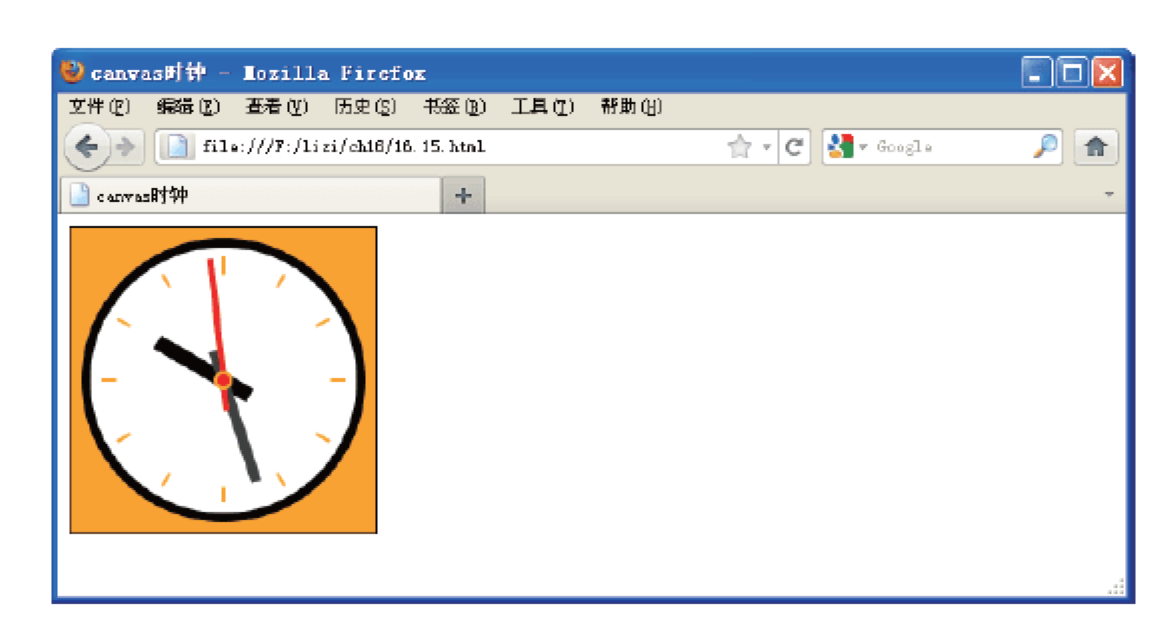


图16-23　时钟特效

02 创建HTML页面。

```
    <html>
    <head>
    <title>canvas时钟</title>
    </head><body>
    <canvas id="canvas" width="200" height="200" style="border:1px solid
#000;">您的浏览器不支持Canvas。</canvas></body></html>
```

上面代码创建了一个画布，其宽度为200px，高度为200px，带有边框，颜色为黑色，样式为直线型。在Firefox 5.0中浏览效果如图16-24所示，可以看到显示了一个带有黑色边框的画布，画布中没有任何信息。

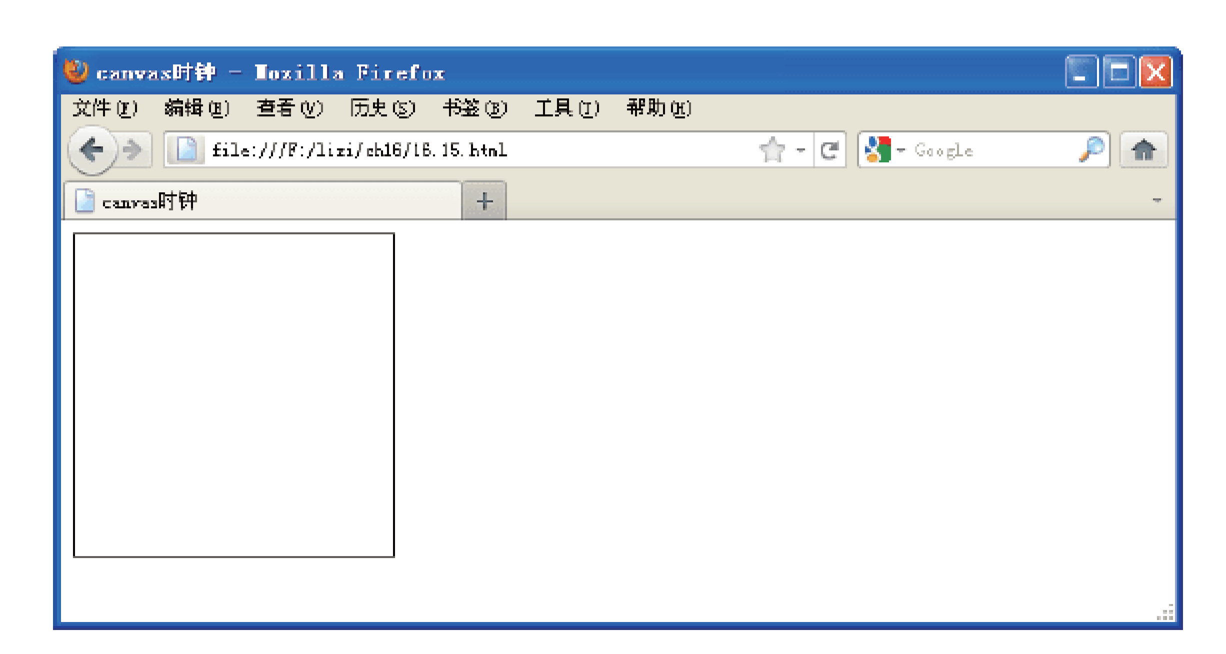


图16-24　定义画布

03 添加JavaScript，绘制不同图形。

```
    <script type="text/javascript" language="javascript" charset="utf-8">
     var canvas = document.getElementById('canvas');
     var ctx = canvas.getContext('2d');
     if(ctx){
      var timerId;
```

```
      var frameRate = 60;
      function canvObject(){
       this.x = 0;
       this.y = 0;
       this.rotation = 0;
       this.borderWidth = 2;
       this.borderColor = '#000000';
       this.fill = false;
       this.fillColor = '#ff0000';
       this.update = function(){
        if(!this.ctx)throw new Error('你没有指定ctx对象。');
        var ctx = this.ctx
        ctx.save();
        ctx.lineWidth = this.borderWidth;
        ctx.strokeStyle = this.borderColor;
        ctx.fillStyle = this.fillColor;
        ctx.translate(this.x, this.y);
        if(this.rotation)ctx.rotate(this.rotation * Math.PI/180);
        if(this.draw)this.draw(ctx);
        if(this.fill)ctx.fill();
        ctx.stroke();
        ctx.restore();
       }  };…
      timerId = setInterval(function(){
       // 清除画布
       ctx.clearRect(0,0,200,200);
       // 填充背景色
       ctx.fillStyle = 'orange';
       ctx.fillRect(0,0,200,200);
       // 表盘
       circle.update();
       // 刻度
       for(var i=0;cache=ls[i++];)cache.update();
       // 时针
       hour.rotation = (new Date()).getHours() * 30;
       hour.update();
       // 分针
       minute.rotation = (new Date()).getMinutes() * 6;
       minute.update();
       // 秒针
       seconds.rotation = (new Date()).getSeconds() * 6;
       seconds.update();
       // 中心圆
       center.update();
      },(1000/frameRate)|0);
     }else{
      alert('您的浏览器不支持Canvas无法预览.\n跟我一起说："很遗憾!"');
     }
    </script>
```

上面代码由于篇幅比较长，只给出了部分代码，完整代码可以在光盘中查询。上面代码

首先绘制不同类型的图形，例如时针，秒针和分针等。然后再将其组合在一起，并根据时间定义时针等指向。在Firefox 5.0中，浏览效果如图16-22所示，可以看到页面中出现了一个时钟，其秒针在不停地移动。

 16.12 综合实例10——颜色选择器

在页面中定义背景色和字体颜色，是比较常见的一种操作，往往选取颜色时比较为难，不知道哪种颜色适合，并且颜色值还不知道是什么。此时可以利用颜色选择器来定义颜色并获取颜色值。本实例将创建一个颜色选择器，以方便获取颜色值，具体步骤如下所示。

01 分析需求。

本实例原理非常简单，就是将几个常用的颜色值进行组合，组合在一起后合并，即是所要选择的颜色值。这些都是利用JavaScript代码完成的。实例的效果如图16-25所示。

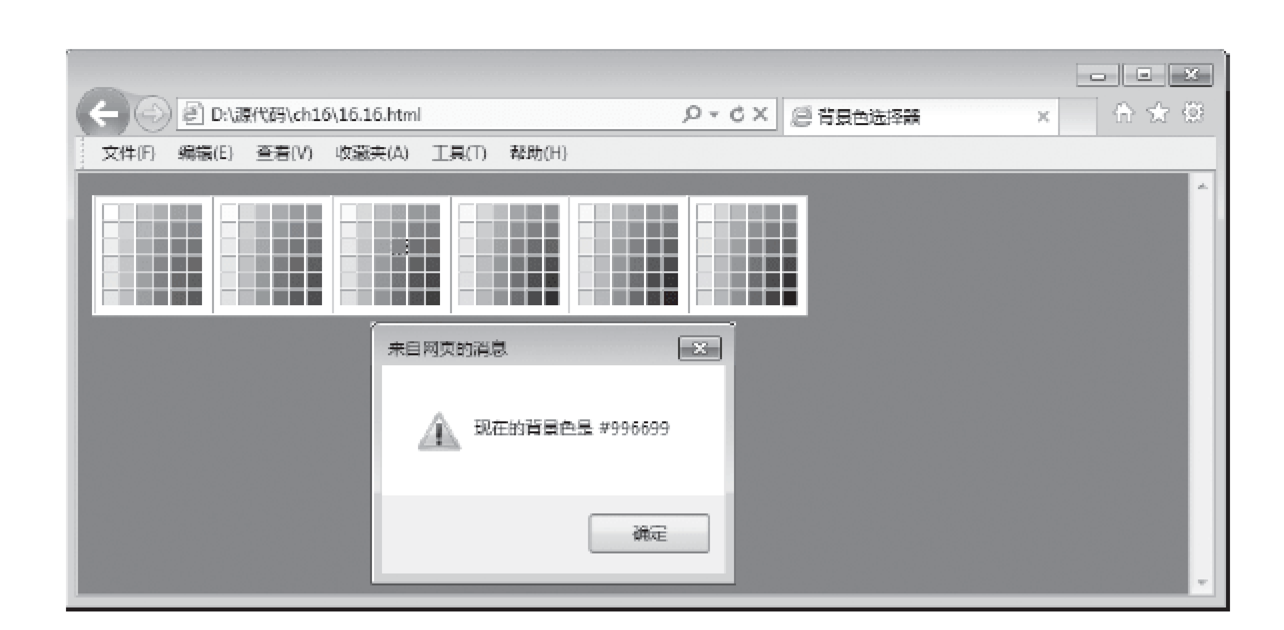


图16-25 设定页面背景色

02 创建基本HTML页面。

```
<html>
<head><title>背景色选择器</title>
</head>
<body bgcolor="#FFFFFF">
</body></html>
```

上述代码比较简单，只实现了一个页面框架，这里就不再分析了。

03 添加JavaScript代码，实现颜色选择。

```
<script language="JavaScript">
<!--
var hex = new Array(6)
hex[0] = "FF"
hex[1] = "CC"
hex[2] = "99"
hex[3] = "66"
```

```
    hex[4] = "33"
    hex[5] = "00"
    function display(triplet)
    {
      document.bgColor = '#' + triplet
      alert('现在的背景色是 #'+triplet)
    }
    function drawCell(red, green, blue)
    {
      document.write('<TD BGCOLOR="#' + red + green + blue + '">')
      document.write('<A HREF="javascript:display(\'' + (red + green + blue) +
'\')">')
      document.write('<IMG SRC="place.gif" BORDER=0 HEIGHT=12 WIDTH=12>')
      document.write('</A>')
      document.write('</TD>')
    }
    function drawRow(red, blue)
    {
      document.write('<TR>')
      for (var i = 0; i < 6; ++i)
      {
        drawCell(red, hex[i], blue)
      }  document.write('</TR>')
    }function drawTable(blue)
    {
      document.write('<TABLE CELLPADDING=0 CELLSPACING=0 BORDER=0>')
      for (var i = 0; i < 6; ++i)
      {
        drawRow(hex[i], blue)
      }
      document.write('</TABLE>')
    }
    function drawCube()
    {
      document.write('<TABLE CELLPADDING=5 CELLSPACING=0 BORDER=1><TR>')
      for (var i = 0; i < 6; ++i)
      {
        document.write('<TD BGCOLOR="#FFFFFF">')
        drawTable(hex[i])
        document.write('</TD>')
      }  document.write('</TR></TABLE>')
    }drawCube()
    // --></script>
```

上面代码中，创建了一个数组对象hex用来存放不同的颜色值。下面几个函数分别将数组中的颜色组合在一起，并在页面中显示，display函数完成定义背景颜色和显示颜色值。

在IE9.0中，浏览效果如图16-25所示，可以看到页面显示多个表格，每个单元格代表一种颜色。



16.13 专家解惑

1．JavaScript中，数组常用的方法有哪些？

（1）join()，将array中的所有element以string的形式连在一起。

```
var a = [1,2,3];
s = a.join();          // s == "1,2,3"
s = a.join(": ");      // s == "1: 2: 3"
```

（2）reverse()，将Array的element顺数颠倒。

```
var a = [1,2,3];
a.reverse();
s = a.join();                 // s == "3,2,1"
```

（3）sort()，排序，默认按字母顺序排序case sensitive，可以自定义排序方式。

```
var a = [111,4,33,2];
a.sort();  // a == [111,2,33,4]
a.sort(function(a,b) {      // a == [2,4,33,111]
     return a-b;              // Returns < 0, 0, or > 0
  });
```

（4）concat()，连接多个Array。

```
var a = [1,2,3];
a.concat(4,5);                // return [1,2,3,4,5]
a.concat([4,5]);              // return [1,2,3,4,5]
a.concat([4,5], [6,7])        // return [1,2,3,4,5,6,7]
a.concat(4,[5,[6,7]]);        // return [1,2,3,4,5,6,7]
```

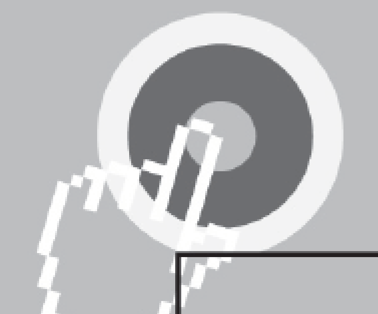
2．在JavaScript中，添加注释有哪些功能？

在程序中，通过添加注释来描述代码的功能，还可以使一些代码无效，以实现逐行检查程序，及时发现并解决问题。JavaScript主要提供了三种注释方法：

（1）单行注释。在需要注释的代码前添加字符“//”，“//”后面的部分会被注释。

（2）多行注释。在代码前添加“/*”，之后添加“*/”，星号之间的部分会被注释。

（3）HTML注释。使用传统的HTML注释，<!--和-->之间的部分会被注释，注释内容可以是一行或多行。



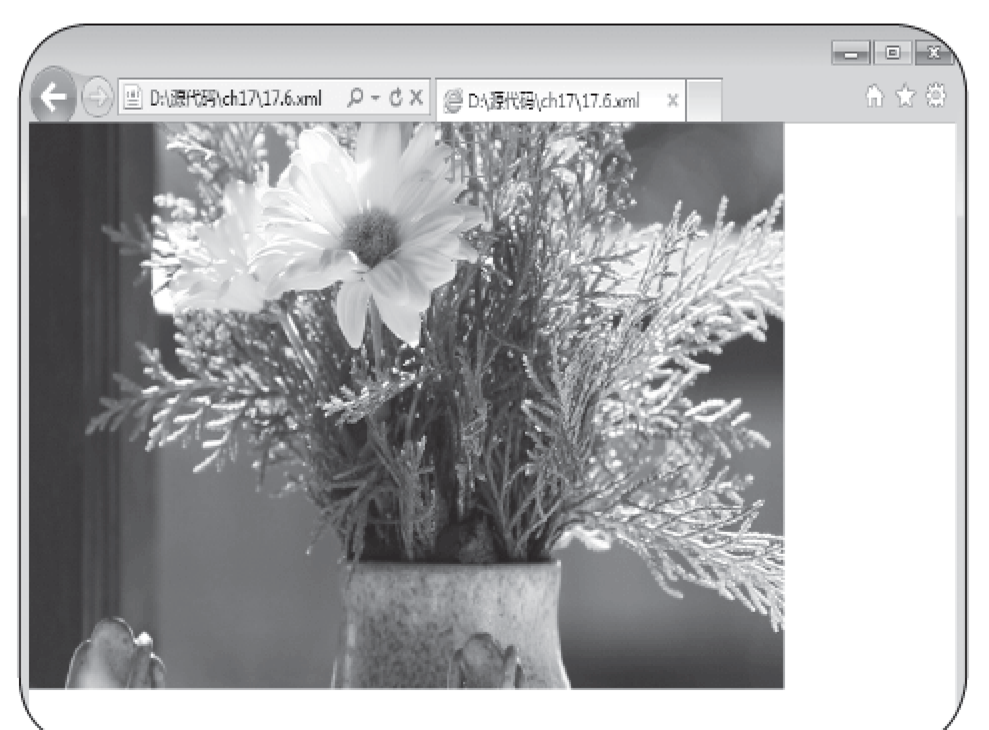
第 17 章

CSS3与XML的综合运用

本章引言

XML是一种标准化的文本格式，可以在Web上表示结构化信息，利用它可以存储有复杂结构的数据信息。XML是HTML的补充，但XML并不是HTML的替代品。在将来的网页开发中，XML将被用来描述、存储数据，而HTML则是用来格式化和显示数据的。







17.1 XML语法基础

XML是标记语言，可支持开发者为Web信息设计自己的标记。XML要比HTML强大得多，它不再是固定的标记，而是允许定义数量不限的标记来描述文档中的资料，允许嵌套的信息结构。

17.1.1 XML概念

随着Internet的发展，为了控制网页显示样式，就增加了一些描述如何显现数据的标记，例如<center>、<b>等标记。但随着HTML的不断发展，W3C组织意识到HTML存在着一些无法避免的问题：

- 不能解决所有解释数据的问题，例如影音文件或化学公式、音乐符号等其他型态的内容。
- 效能问题，需要下载整份文件，才能开始对文件做搜寻的动作。
- 扩充性、弹性、易读性均不佳。

为了解决以上问题，专家们使用SGML精简制作，并依照HTML的发展经验，产生出一套上规则严谨，但是使用简单的描述数据语言：XML。

XML（eXtensible Markup Language，可扩展标记语言）是W3C推荐的参考通用标记语言，同样也是SGML的子类，可以定义自己的一组标记。它具有下面几个特点：

(1) XML是一种元标记语言，所谓“元标记语言”就是开发者可以根据需要定义自己的标记，例如开发者可以定义标记<book>、<name>等，任何满足XML命名规则的名称都可以作为标记，这就为不同的应用程序的应用打开了大门。

(2) 允许通过使用自定义格式、标识，交换和处理数据库可以理解的数据。

(3) 基于文本的格式，允许开发人员描述结构化数据并在各种应用之间发送和交换这些数据。

(4) 有助于在服务器之间传输结构化数据。

(5) XML使用的是非专有的格式，不受版权、专利、商业秘密或是其他种类的知识产权的限制。XML的功能是非常强大的，同时对于人类或是计算机程序来说，都容易阅读和编写，因而成为交换语言的首选。网络带给人类的最大好处是信息共享，在不同的计算机发送数据，而XML是用来告诉我们“数据是什么”，利用XML可以在网络上交换任何一种信息。

【例17.1】实例文件：ch17\17.1.xml

```
<?xml version="1.0" encoding="GB2312" ?>
<电器>
    <家用电器>
        <品牌>小天鹅洗衣机</品牌>
        <购买时间>2009-10-01</购买时间>
        <价格  币种="人民币">899元</价格>
```

```
    </家用电器>
      <家用电器>
        <品牌>海尔冰箱</品牌>
        <购买时间>2011-08-16</购买时间>
        <价格  币种="人民币">3990</价格>
        </家用电器>
</电器>
```

此处需要将文件保存为XML文件。该文件中，每个标记都是用汉语编写的，是自定义标记。整个电器可以看作是一个对象，该对象包含多个家用电器，家用电器是用来存储电器的相关信息的，也可以说家用电器对象是一种数据结构模型。在页面中没有对那个数据的样式进行修饰，而只告诉我们数据结构是什么，数据是什么。

在IE9.0中浏览效果如图17-1所示，可以看到整个页面树形结构显示，通过单击“-”可以关闭整个树形结构，单击“+”可以展开树形结构。

图17-1 XML文件显示

17.1.2 XML文档组成和声明

一个完整的XML文档由声明、元素、注释、字符引用和处理指令组成。在文档中，所有这些XML文档的组成部分都是通过元素标记来指明的。可以将XML文档分为三个部分，如图17-2所示。

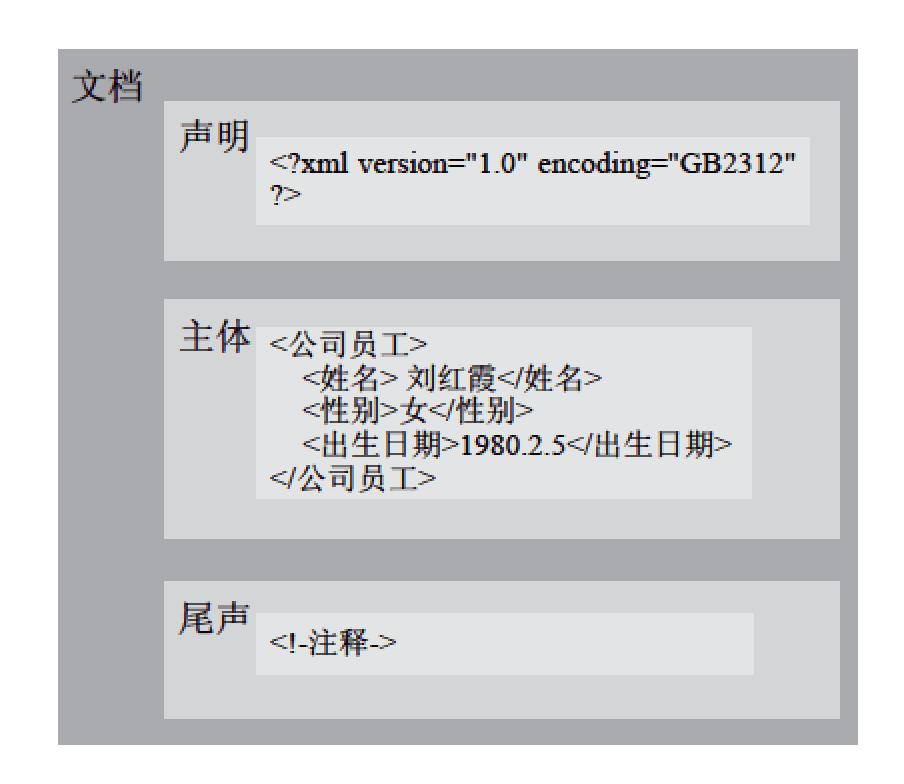


图17-2 XML文档组成

XML声明必须作为XML文档的第一行，前面不能有空白、注释或其他处理指令。完整的声明格式如下：

```
<?xml version="1.0" encoding="编码" standalone="yes/no" ?>
```

其中version属性不能省略，且在属性列表中必须排在第一位，指明所采用的XML的版本号，值为1.0。该属性用来保证对XML未来版本的支持。encoding属性是可选属性。该属性指定了文档采用的编码方式，即规定了采用哪种字符集对XML文档进行字符编码，常用的编码方式为UTF-8和GB2312。如果没有使用encoding属性，那么该属性的默认值是UTF-8；如果encoding属性值设置为GB2312，则文档必须使用ANSI编码保存，文档的标记以及标记内容只可以使用ASCII字符和中文。

使用GB2312编码的XML声明如下：

```
<?xml version="1.0" encoding="GB2312" ?>
```

XML文档主体必须有根元素，所有的XML必须包含可定义根元素的单一标记对，所有其他的元素都必须处在这个根元素内部。所有的元素均可拥有子元素，子元素必须被正确地嵌套在它们的父元素内部。根标记以及根标记内容共同构成XML文档主体，没有文档主体的XML文档将不会被浏览器或其他XML处理程序识别。

注释可以提高文档的阅读性，尽管XML解析器通常会忽略文档中的注释，但位置适当且有意义的注释可以大大提高文档的可读性，所以XML文档中不用于描述数据的内容都可以包含在注释中。注释以“<!--”开始，以“-->”结束，在起始符和结束符之间的内容为注释内容，注释内容可以输入符合注释规则的任何字符串。

【例17.2】实例文件：ch17\17.2.xml

```
<?xml version="1.0" encoding="gb2312"?>
<!--这是一个优秀学生名单-->
<学生名单>
<学生>
  <姓名>刘海松</姓名>
    <学号>21</学号>
    <性别>男</性别>
</学生>
<学生>
  <姓名>刘红霞</姓名>
    <学号>22</学号>
      <性别>女</性别>
</学生>
</学生名单>
```

上面代码中，第一句代码是一个XML声明。“<学生>”标记是“<学生名单>”标记的子元素，而“<姓名>”标记和“<学号>”标记是“<学生>”的子元素。“<!---->”是一个注释。

在IE9.0中浏览效果如图17-3所示，可以看到页面显示了一个树形结构，并且数据层次感非常好。

```
D:\源代码\ch17\17.2.xml

<?xml version="1.0" encoding="GB2312"?>
<!--这是一个优秀学生名单-->
- <学生名单>
  - <学生>
      <姓名>刘海松</姓名>
      <学号>21</学号>
      <性别>男</性别>
    </学生>
  - <学生>
      <姓名>刘红霞</姓名>
      <学号>22</学号>
      <性别>女</性别>
    </学生>
  </学生名单>
```

图17-3 XML文档组成

17.1.3 XML元素介绍

元素是以树形分层结构排列的，它可以嵌套在其他元素中。

1. 元素类别

在XML文档中，元素也分为非空元素和空元素两种类型。一个XML非空元素是由开始标记、结束标记以及标记之间的数据构成的。开始标记和结束标记用来描述标记之间的数据，标记之间的数据被认为是元素的值。非空元素的语法结构如下：

```
<开始标记>文本内容</结束标记>
```

而空元素就是不包含任何内容的元素，即开始标记和结束标记之间没有任何内容的元素。其语法结构如下：

```
<开始标记></结束标记>
```

可以把元素内容为文本的非空元素转换为空元素。例如，

```
<hello>下午好</hello>
```

<hello>是一个非空元素，如果把非空元素的文本内容转换为空元素的属性，那么转换后的空元素可以写为：

```
<hello content="下午好"></hello>
```

2. 元素命名规范

XML元素命名规则与Java、C等命名规则类似，它也是一种对大小写敏感的语言。XML元素命名必须遵守下列规则：

（1）元素名中可以包含字母、数字和其他字符，如<place>、<地点>、<no123>等。元素名中虽然可以包含中文，但是在不支持中文的环境中将不能够解释包含中文字符的XML文档。

（2）元素名中不能以数字或标点符号开头。例如<123no>、<.name>、<?error>元素名称都是非法名称。

（3）元素名中不能包含空格，如<no 123>是非法名称。

3．元素嵌套

元素的内容可以包含子元素。子元素本身也是元素，被嵌套在上层元素之内。如果子元素嵌套了其他元素，那么它同时也是父元素，例如下面所示部分代码：

```
<?xml version="1.0" encoding="gb2312" ?>
<students>
  <student>
    <name>张三</name>
    <age>20</age>
  </student>
  ...
</students>
```

<student>是<students>的子元素，同时也是<name>和<age>的父元素，而<name>和<age>是<student>的子元素。

4．元素实例

【例17.3】实例文件：ch17\17.3.xml

```
<?xml version="1.0" encoding="gb2312" ?>
<通讯录>
  <!--"记录"标记中包含姓名、地址、电话和电子邮件 -->
  <记录 date="2011/2/1">
    <姓名>刘海松</姓名>
    <地址>河南省郑州市中州大道</地址>
    <电话>0371-12345678</电话>
    <电子邮件>evan@tom.com</电子邮件>
  </记录>
  <记录 date="2011/3/12">
    <姓名>刘红霞</姓名>
    <地址>河北省邯郸市工农大道2号</地址>
    <电话>83365354</电话>
  </记录>
  <记录 date="2011/6/23">
    <姓名>闫阳</姓名>
    <地址>吉林省长春市幸福路6号</地址>
    <电话>0431-8365257</电话>
    <电子邮件>yanyang@sina.com</电子邮件>
  </记录>
</通讯录>
```

文件代码中，第一行是XML声明，它声明该文档是XML文档，文档所采用的版本号以及文档使用的字符编码集。在这个例子中，遵守的是XML 1.0版本规范，字符编码是GB2312编码方式。<记录>是<通讯录>的子标记，但<记录>标记同时是<姓名>和<地址>等标记的父元素。

在IE9.0中浏览效果如图17-4所示，可以看到页面显示了一个树形结构，每个标记中间包含相应的数据。

```
D:\源代码\ch17\17.3.xml

<?xml version="1.0" encoding="GB2312"?>
- <通讯录>
      <!--"记录"标记中包含姓名、地址、电话和电子邮件 -->
  - <记录 date="2011/2/1">
      <姓名>刘海松</姓名>
      <地址>河南省郑州市中州大道</地址>
      <电话>0371-12345678</电话>
      <电子邮件>evan@tom.com</电子邮件>
    </记录>
  - <记录 date="2011/3/12">
      <姓名>刘红霞</姓名>
      <地址>河北省邯郸市工农大道2号</地址>
      <电话>83365354</电话>
    </记录>
  - <记录 date="2011/6/23">
      <姓名>闫阳</姓名>
      <地址>吉林省长春市幸福路6号</地址>
      <电话>0431-8365257</电话>
      <电子邮件>yanyang@sina.com</电子邮件>
    </记录>
</通讯录>
```

图17-4 元素包含



17.2 CSS修饰XML文件

我们知道，XML文档本身只包含数据，而没有关于显示数据样式的信息。如果需要将XML文档数据美观的显示出来，而不是以树形结构显示，则可以通过CSS来控制XML文档中各个元素的呈现方式。

17.2.1 XML使用CSS

XML文档数据需要使用CSS属性定义显示样式，其方法是把CSS代码做成独立文件，然后引入到XML中。在XML文档引入样式表CSS，可以将数据的内容和表示分离出来，并且能够实现CSS的重复使用。

XML文档中引用CSS文件，在XML文件中必须使用下面的操作指令：

```
<?xml-stylesheet href="URI" type="text/css"?>
```

xml-stylesheet表示在这里使用了样式表。样式表的URI是表示要引入文件所在的路径，如果只是一个文件的名字，该CSS文件必须和XML文档同在一个目录的下面；如果URI是一个链接，该链接必须是有效的和可访问的。type表示该文件所属的类型是文本形式，其内容是CSS代码。

【例17.4】实例文件：ch17\17.4.xml

```
<?xml version="1.0" encoding="GB2312" ?>
<?xml-stylesheet type="text/css" href="17.4.css"?>
```

```
<student>
<name>孙福全</name>
<sex>男</sex>
<name>王小玲</name>
<sex>女</sex>
</student>
```

实例文件：ch17\17.4.css

```
student{
background-color: #ddeecc;
font-family:"幼圆";
text-align:center;
display:block;
}
name{
font-size:20px;
color:red;
}
sex{
font-size:12px;
font-style:italic;
}
```

在CSS文件中，针对student、name和sex 3个标记，设置了不同的显示样式，例如字体大小、字体颜色、对齐方式等。

在IE9.0中，浏览效果如图17-5所示，可以看到XML文档不再是以树形结构显示，并且没有标记出现，而只是显示其标记中的数据。

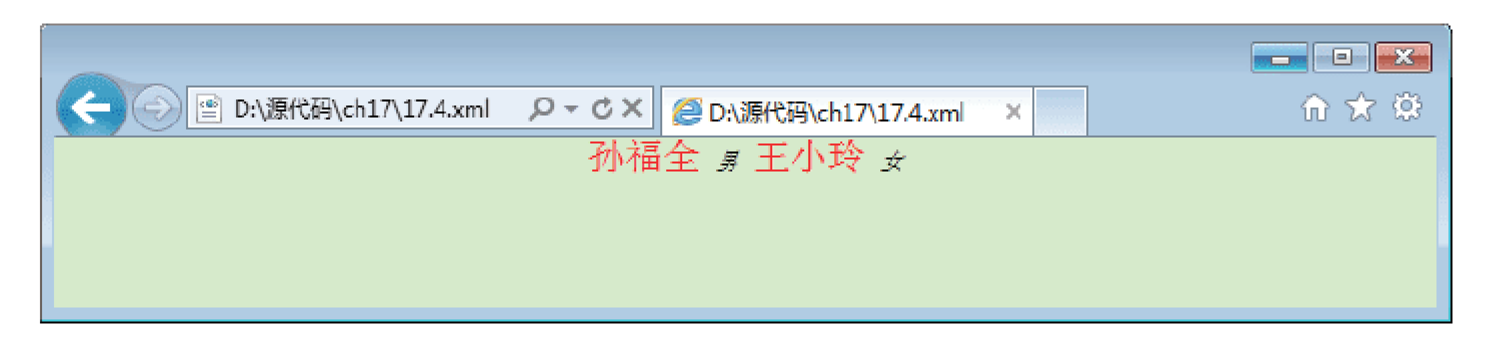


图17-5 XML引入CSS文件

17.2.2 设置字体属性

CSS3样式表提供了多种字体属性，使设计者对字体有更详细的设置，从而能够使页面效果更加丰富。例如font-style、font-variant、font-weight、font-size和font-family等属性，这些属性前面已经介绍过，就不再重复了。这些字体属性，同样可以应用于XML文件元素。

【例17.5】实例文件：ch17\17.5.xml

```
<?xml version="1.0" encoding="GB2312"?>
<?xml-stylesheet href="17.5.css" type="text/css"?>
<company>
  <name>水月网页设计工作室</name>
  <address>郑州市花园路松风大夏</address>
```

```
    <phone>13012345678</phone>
</company>
```

实例文件：ch17\17.5.css

```
company{
  color: #ddeecc;
  font:normal small-caps bolder 15pt "幼圆" ;
               background-color:#123543
}
name{
   font-size:30px;
   display:block;
}
address{
              font-size: 12px;
  display:block;
}
phone{
            font-size: 12px;
            font-style:italic;
            display:block;
}
```

在上面的CSS代码中，针对XML中的标记，进行了字体、背景颜色和前景色设置。

在IE9.0中，浏览效果如图17-6所示，可以看到网页显示了一个公司介绍信息，其中字体大小不一样，联系方式以斜体显示。

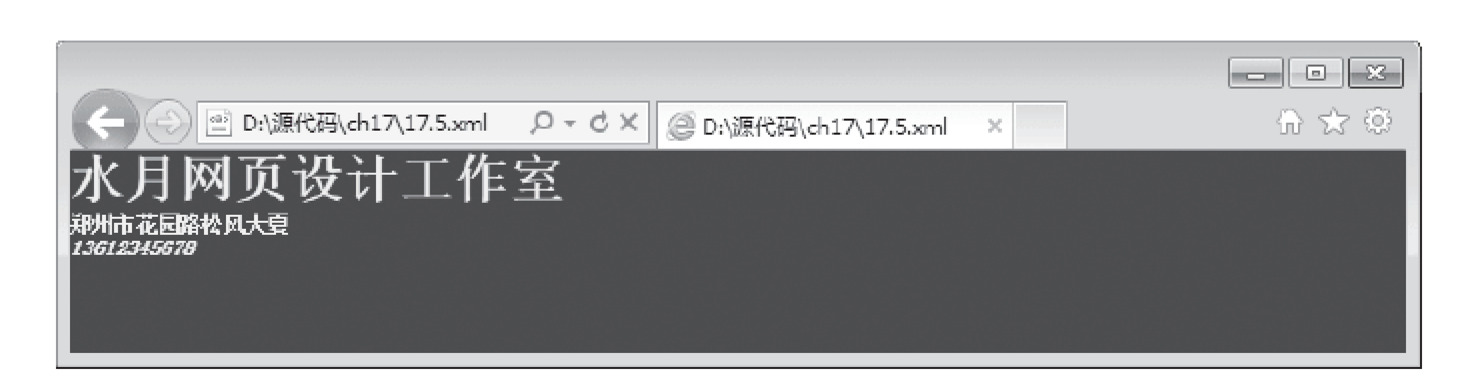


图17-6 CSS定义XML字体属性

17.2.3 设置色彩属性

颜色和背景是网页设计时两个重要的因素，一个颜色搭配协调、背景优美的文档总是能吸引不少的访问者。CSS的强大表现功能在控制XML元素的颜色和背景方面同样发挥得淋漓尽致。XML元素的背景可设置成一种颜色或一幅影像。

在CSS3中，如果需要设置文本颜色，即网页前景色，通常使用color属性，定义元素背景，其相关属性分别如下：background-color、background-image、background-repeat、background-attachment、background-position。这些前面都已经介绍过，这里就不再介绍了。

【例17.6】实例文件：ch17\17.6.xml

```
<?xml version="1.0" encoding="GB2312" ?>
<?xml-stylesheet href="17.6.css" type="text/css" ?>
```

```
<img>
插花
</img>
```

实例文件：ch17\17.6.css

```
img{
  display:block;
  color:red;
  text-align:center;
  font-size:40px;
  left:50px;
  top:170px;
  background-image:URL("08.jpg");
  background-repeat:no-repeat;
  background-position:left;
}
```

上面CSS代码设置背景以块显示，字体颜色为红色，字体大小为40px并居中显示。background-image引入背景图片为08.jpg，并设置了图片不重复。

在IE9.0中，浏览效果如图17-7所示，可以看到页面背景为一张图片，且不重复，在图片上显示了“插花”两个红色字体。



图17-7　CSS定义XML背景

17.2.4　设置边框属性

在CSS3中，可以使用border-style、border-width和border-color这3个属性设定边框。页面元素的边框就是将元素内容及间隙包含在其中的边线，类似于表格的外边线。页面元素边框通过三个方面来描述：宽度、样式和颜色，这三个方面决定了边框显示出来的外观。

【例17.7】实例文件：ch17\17.7.xml

```
<?xml version="1.0" encoding="GB2312" ?>
<?xml-stylesheet href="17.7.css" type="text/css" ?>
<border>
```

```
        <smallBorder>
          学校雷锋好榜样
      </smallBorder>
</border>
```

实例文件：ch17\17.7.css

```
border{
    border-style:solid;
    border-width:15px;
    border-color:#123456;
    width:200px;
    height:150px;
     text-align:center;
}
smallborder{
 font-size:20px;
 color:red;
}
```

在border标记中，设置边框显示样式，例如直线显示，颜色为深蓝色，宽度为15px，并且设置显示块的宽度为200px，高度为150px，边框内元素居中显示。在smallborder标记中设置了字体大小和字体颜色。

在IE9.0中，浏览效果如图17-8所示，可以看到页面中显示了一个边框，边框中显示的是红色字体，其内容是“学习雷锋好榜样”。



图17-8 设置XML元素边框

17.2.5 设置文本属性

在CSS3中，提供多种文本属性来实现对文本的控制，例如text-indent、text-align、white-space、line-height、vertical-align、text-transform和text-decoration等属性。这些属性前面已经介绍过，这里就不再介绍了。利用上面这些属性，可以控制XML元素的显示样式。

【例17.8】实例文件：ch17\17.8.xml

```
<?xml version="1.0" encoding="gb2312"?>
<?xml-stylesheet type="text/css" href="17.8.css"?>
```

```
    <big>
      <one>健康</one>
    <two>
    <title>饮茶养生养颜 特殊时期慎饮茶</title>
    <content>
    金银花，味甘，性寒，具有清热解毒、疏散风热的作用。金银花为清热解毒之良药，既能清里热，
又能散表热，临床上主要用于治疗各种痈肿疮毒、热毒血痢及温热病等。金银花药性偏寒，不适合长期饮
用，仅适合在炎热的夏季暂时饮用以防治痢疾。
    </content>
    </two>
    </big>
```

实例文件：ch17\17.8.css

```
    big{
      width:500px;
      border:#6600FF 1px solid;
      height:200px;
      font-size:12px;
      font-family:"幼圆";

      }
    one{
      font-size:18px;
      width:500px;
      height:25px;
      line-height:25px;
      text-align:center;
      color:#FF3300;
      margin-top:5px;
      font-weight:800;
                  text-decoration:underline;
      }
    title{
      margin:10px 0 10px 10px;
      display:block;
      color:#0033FF;
      font-size:14px;
      font-weight:800;
    text-align:center;
      }
    content{
      display:block;
      line-height:20px;
      width:490px;
      margin-left:10px;
      font-weight:800;
      text-indent:2em;
      }
```

上面CSS代码分别定义不同标记的显示样式，例如宽度、高度、边框样式、字体大小、行高和是否带有下划线等。

在IE9.0中，浏览效果如图17-9所示，可以看到页面中显示了一个公告栏，公告栏中显示了不同颜色的字体，并且段落缩进两个单元格显示。

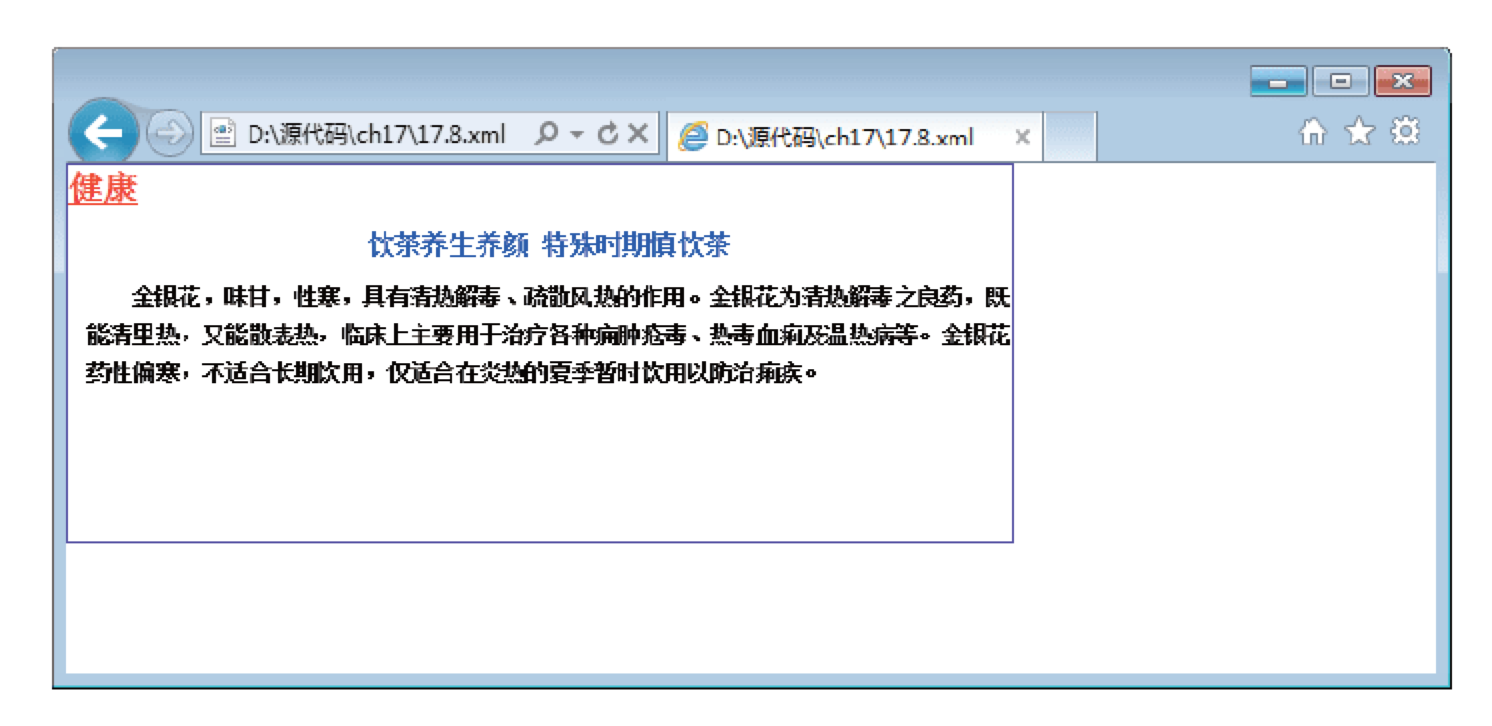


图17-9 修饰XML文本

 17.3 实例1：招聘广告

CSS3结合XML文档，可以创建出多种多样的样式，例如在HTML页面中常见的招聘信息。本实例将结合前面学习的XML和CSS知识，创建一个基于XML文档的招聘广告。具体步骤如下所示。

01 分析需求。

创建一个招聘广告，就文件而言包括两大部分，一个是XML文档，另一个是CSS文件。XML文档需要包含招聘标题、招聘内容和联系方式，而CSS文件需要针对XML文档标记进行样式设计。实例效果如图17-10所示。

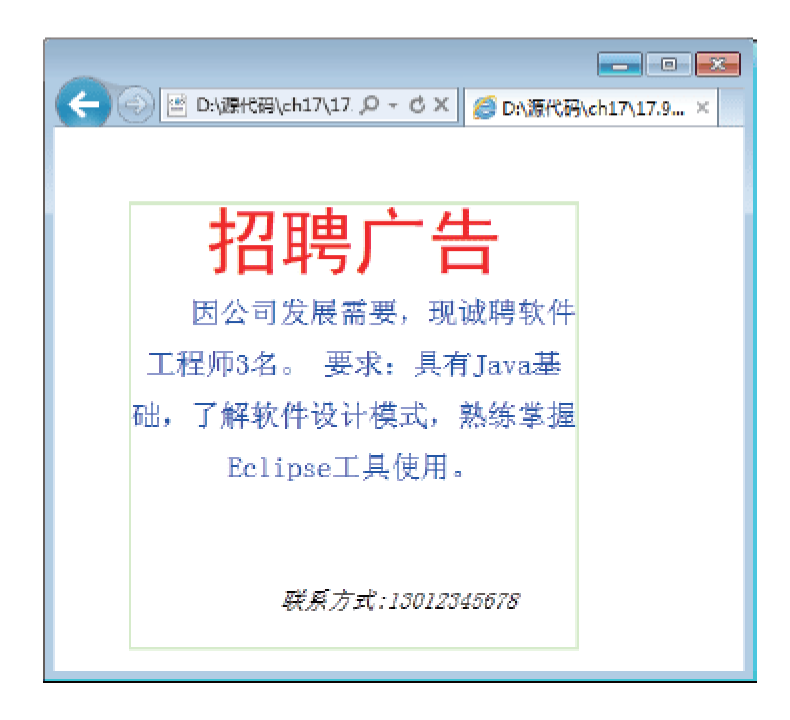


图17-10 招聘显示

02 构建XML文档。

```
<?xml version="1.0" encoding="GB2312" ?>
<?xml-stylesheet href="17.9.css" type="text/css" ?>
```

```
<zhaopin>
   <title>
       招聘广告
   </title>
   <content>
      因公司发展需要，现诚聘软件工程师3名。
         要求：具有Java基础，了解软件设计模式，熟练掌握Eclipse工具使用。
   </content>
   <address>
       联系方式:13012345678
   </address>
</zhaopin>
```

在XML文件中，有一个根标记<zhaopin>，下面包含了三个子标记，分别为<title>标记、<content>标记和<address>标记。标记中分别存放着相应的数据。

在IE9.0中，浏览效果如图17-11所示，可以看到页面中显示了一个树形结构，其元素中包含不同的信息。

03 添加CSS样式文件，修饰整体样式。

```
zhaopin{
  display:block;
  position:absolute;
  top:50px;
  left:50px;
  width:300px;
  height:300px;
  border:2px #ddeecc solid;
  text-align:center;
}
```

上面代码定义了zhaopin元素以块形式显示，绝对定位，坐标为（50,50），高度和宽度都是300px。

在IE9.0中，浏览效果如图17-12所示，可以看到页面不再以树形结构显示，而是显示了一个边框，边框内显示的是XML文档的数据。



图17-11 XML文档显示

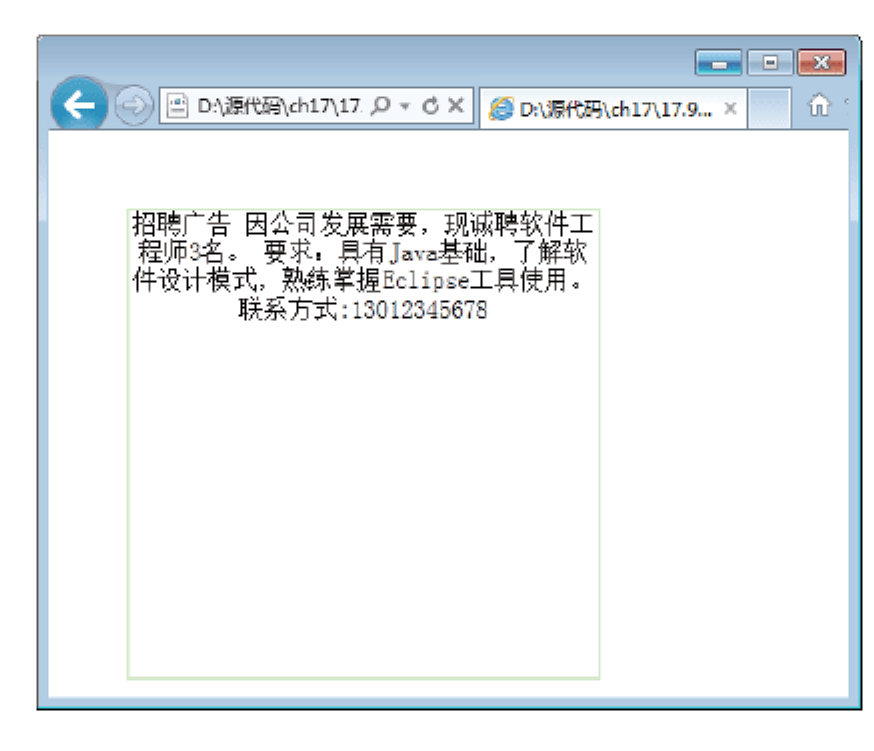


图17-12 定义边框样式

04 添加CSS样式，修饰标题、内容和地址信息。

```
title{
   font-family:黑体;
   font-varanit:normal;
   font-size:50px;
  color:red;
}
content{
   display:block;
   font-family:"幼圆";
   font-varanit:normal;
   font-size:20px;
   color:blue;
   text-indent:2em;
   line-height:35px;
}
address{
  position:absolute;
  left:100px;
   top:260px;
  text-align:right;
  font-style:italic;
}
```

在CSS文件中，针对上面不同的标记，设置相应的样式，例如修饰字体、定位和块的大小。

在IE9.0中浏览效果如图17-10所示，可以看到页面边框内，标题以红色字体显示，正文以蓝色字体显示，联系方式以斜体显示。



17.4 实例2：书目清单

在网页中，使用表格显示数据是最常见的一种方式。本实例将使用XML和CSS模拟表格效果，即创建一个书目清单，具体步骤如下所示。

01 分析需求。

创建一个具有表格效果的页面，就是利用元素的边框效果完成的。针对XML标记设定一个边框效果，相近的边框重叠就可以了。实例完成后效果如图17-13所示。

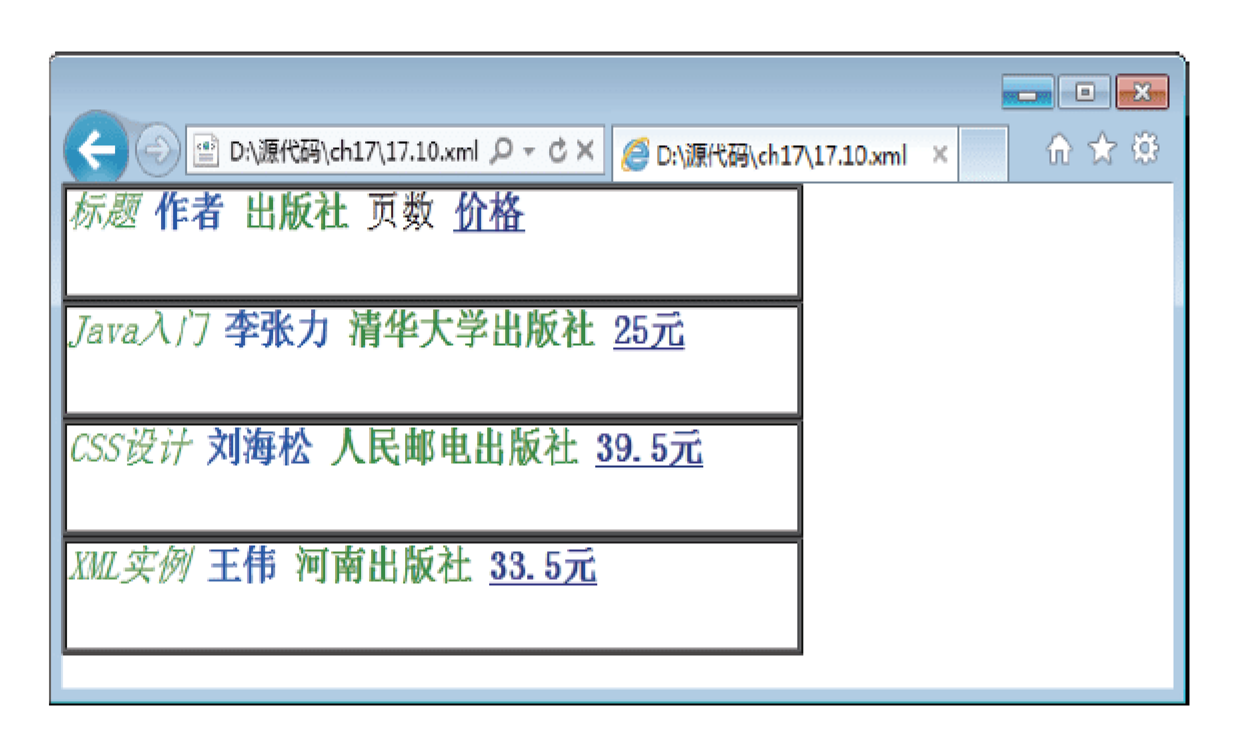


图17-13 模拟表格效果

02 创建XML文档。

```
<?xml version="1.0" encoding="GB2312"?>
<?xml-stylesheet type="text/css" href="17.10.css"?>
<inventory>
     <image></image>
     <book>
        <title>标题</title>
        <author>作者</author>
        <binding>出版社</binding>
        <pages>页数</pages>
        <price>价格</price>
     </book> <book>
        <title>java入门</title>
        <author>李张力</author>
        <binding>清华大学出版社</binding>
        <price>25元</price>
     </book>     <book>
        <title>css设计</title>
        <author>刘海松</author>
        <binding>人民邮电出版社</binding>
        <price>39.5元</price>
     </book>     <book>
         <title>xml实例</title>
         <author>王伟</author>
         <binding>河南出版社</binding>
        <price>33.5元</price>
     </book>
</inventory>
```

在XML文件中的第二行，使用xml-stylesheet标记将CSS文件引入到XML文件中，下面创建了一个book标记及其子标记title、author、price等。每个标记都包含不同的数据。

在IE9.0中浏览效果如图17-14所示，可以看到页面以属性结构显示XML文件。

```
D:\源代码\ch17\17.10.xml
<?xml version="1.0" encoding="GB2312"?>
- <INVENTORY>
    <image/>
  - <BOOK>
      <TITLE>标题</TITLE>
      <AUTHOR>作者</AUTHOR>
      <BINDING>出版社</BINDING>
      <PAGES>页数</PAGES>
      <PRICE>价格</PRICE>
    </BOOK>
  - <BOOK>
      <TITLE>Java入门</TITLE>
      <AUTHOR>李张力</AUTHOR>
      <BINDING>清华大学出版社</BINDING>
      <PRICE>25元</PRICE>
    </BOOK>
  - <BOOK>
      <TITLE>CSS设计</TITLE>
      <AUTHOR>刘海松</AUTHOR>
      <BINDING>人民邮电出版社</BINDING>
      <PRICE>39.5元</PRICE>
    </BOOK>
  - <BOOK>
      <TITLE>XML实例</TITLE>
      <AUTHOR>王伟</AUTHOR>
      <BINDING>河南出版社</BINDING>
      <PRICE>33.5元</PRICE>
    </BOOK>
  </INVENTORY>
```

图17-14 图书目录显示

03 创建CSS文件，修饰指定元素。

```
book{
        display:block;
        font-size:20px;
        text-align:left;
        border-style:ridge;
        color:black;
        width:420px;
         height:50px;
}
title{
        font-style:italic;
        color:#009900;
}
author{
        font-weight:bold;
        color:BLUE;
}
binding{
        font-weight:bold;
        color:GREEN;
}
price{
       font-weight:bold;
       color:#000099;
       text-align: left;
       font-weight: bold;
       text-decoration:underline;
}
```

从代码中，可以看出针对XML文件里的标记，定义了不同的样式。其思路为先定义book标记样式，然后再定义其子标记样式。

在IE9.0中浏览效果如图17-13所示，可以看到页面中以表格方式显示不同的数据，其数据以不同的颜色分开。



17.5 实例3：图文混排页面

图文搭配布局是显示的永恒话题，文字进行介绍，图形进行说明二者相得益彰、互为补充。本实例使用XML文档结合CSS文件完成图文混搭的布局，具体步骤如下所示。

01 分析需求。

将图形和文字放在一起进行布局，需要将图形放在一个指定位置显示，最好定义其显示大小，文字可以环绕图片，也可以放在一侧显示。实例完成后，效果如图17-15所示。

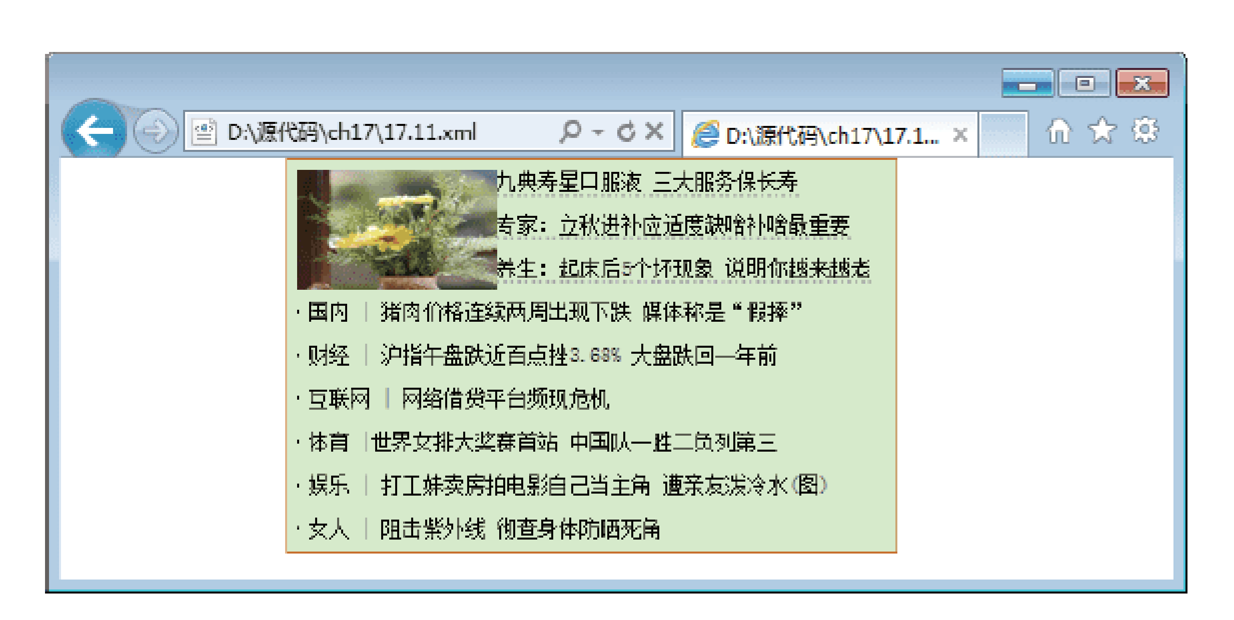


图17-15 图文混排

02 构建XML文档。

```
<?xml version="1.0" encoding="GB2312"?>
<?xml-stylesheet type="text/css" href="17.11.css"?>
<xinwen>
  <content>
    <right>
      <img></img>
      九典寿星口服液 三大服务保长寿
      <br />
      专家：立秋进补应适度缺啥补啥最重要
      <br />
      养生：起床后5个坏现象 说明你越来越老
      <br/>
    </right>
    ·国内 | 猪肉价格连续两周出现下跌 媒体称是“假摔”
    <br />
    ·财经 | 沪指午盘跌近百点挫3.68% 大盘跌回一年前
    <br />
```

```
        ·互联网 | 网络借贷平台频现危机
        <br />
        ·体育 |世界女排大奖赛首站 中国队一胜二负列第三
        <br />
        ·娱乐 | 打工妹卖房拍电影自己当主角 遭亲友泼冷水(图)
        <br />
        ·女人 | 阻击紫外线 彻查身体防晒死角
    </content>
    </xinwen>
```

在XML文件中，首先定义了根标记xinwen，然后定义了content标记，content标记中包含了right、img和br标记。

在IE9.0中浏览效果如图17-16所示，可以看到页面中显示了一个树形结构，其元素中包含了不同的文本数据。

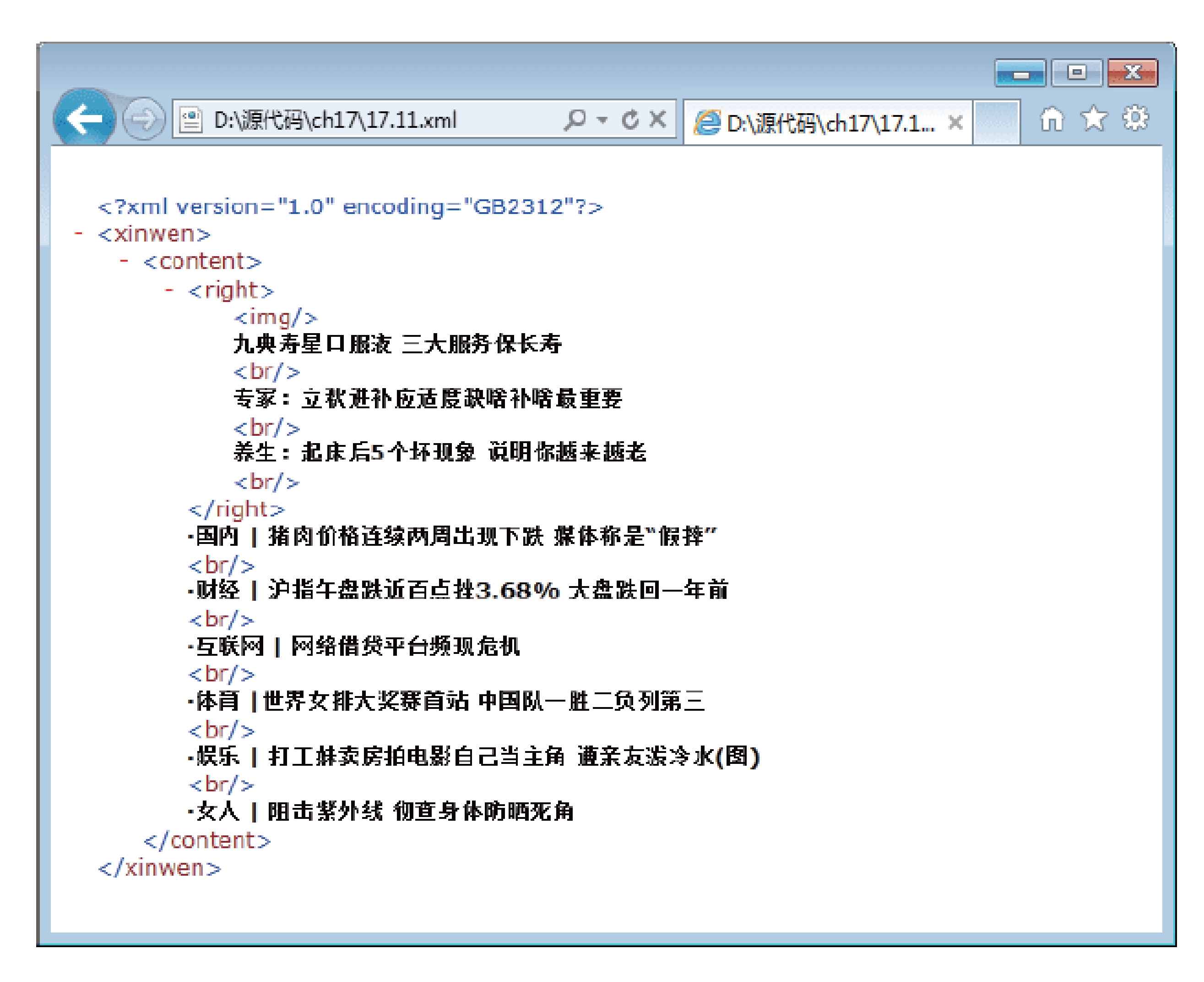


图17-16 图文混排树形结构

03 添加CSS样式，修饰xinwen、br和content元素。

```
    xinwen{
      width:400px;
      font-size:12px;
      font-family:"宋体";
      margin:0 auto 0 auto;
      }
    br{
      display:block;
      }
    content{
                  width:350px;
      border:#CC6600 1px solid;
      float:left;
      line-height:25px;
      margin-left:10px;
```

```
    background-color:#ddeecc;
    }
```

在CSS文件中，创建了不同标记选择器，针对XML文档标记进行修饰。基本都是设置字体、边框、定位和背景属性。需要说的是，由于使用了XML文档，所有
标记不能像在HTML里那样实现换行的效果，这里需要为br定义CSS样式，才能实现正文里的换行效果。

在IE9.0中浏览效果如图17-17所示，可以看到页面中显示了黄色边框，边框内显示了相关列表信息。

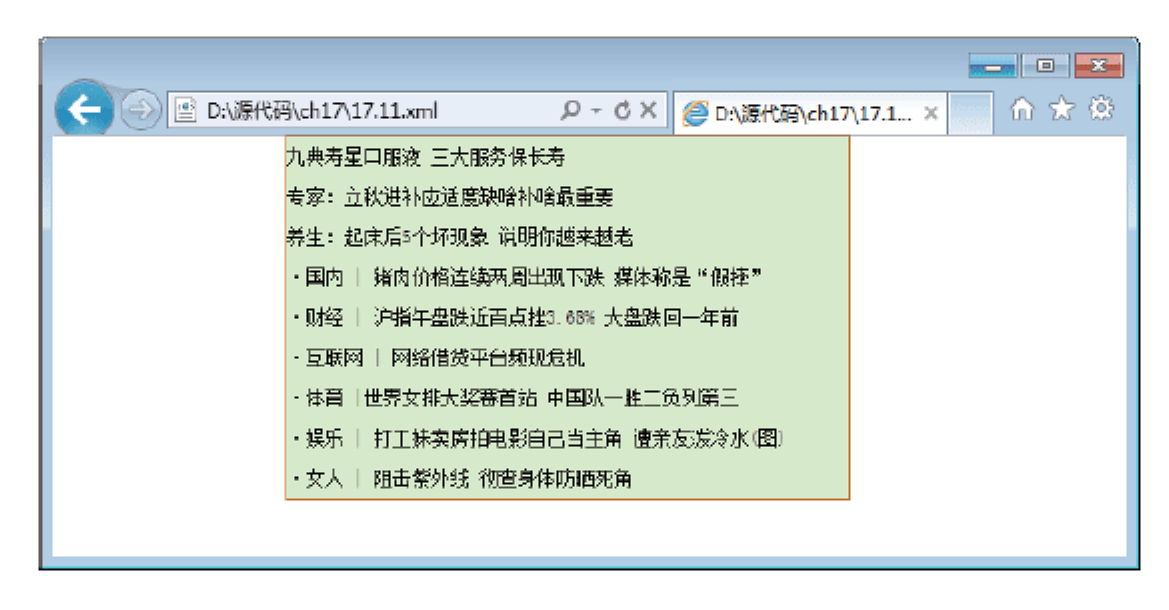


图17-17 修饰指定标记

04 添加CSS样式，修饰img和right元素。

```
img{
  width:115px;
  height:70px;
  background-image:url(003.jpg);
  float:left;
  margin:5px 0 0 5px;
  }
right{
  border-bottom:#999999 dotted 2px;
  }
```

上面的CSS代码，定义了img元素显示样式，例如宽度、高度、背景图片、左浮动和外边距距离等。

在IE9.0中，浏览效果如图17-15所示，可以看到页面边框内显示了一张图片。



17.6 实例4：古诗欣赏

在一些古典风格的网站中，往往有很多漂亮的古诗页面，用来展现诗歌。本实例将模拟这种效果，具体步骤如下所示。

01 分析需求。

如果要对古诗内容进行展示，需要通过绝对定位的方法，为XML文档的段落文字排版。

同样也可以导入一个漂亮的背景，配合文字展示。实例完成后效果如图17-18所示。

02 创建XML文档。

```
<?xml version="1.0" encoding="GB2312"?>
<?xml-stylesheet type="text/css" href="17.12.css"?>
<qiu>  <title>立秋</title>
  <author>宋  刘翰 </author>
  <content>
            乳鸦啼散玉屏空，一枕新凉一扇风。<br/>
            睡起秋色无觅处，满阶梧桐月明中 。<br/>
            </content></qiu>
```

创建XML文档，文档内容非常简单，创建了一个根标记qiu及其子标记title、author和content。xml-stylesheet表示引入一个CSS文件。在IE9.0中浏览效果如图17-19所示，可以看到页面显示了一个树形结构。



图17-18 古诗欣赏

图17-19 古诗树形结构

03 创建CSS文件，修饰qiu 元素。

```
qiu{
  margin:0px;
  background:url(lan.jpg) no-repeat;  /*添加背景图片 */
  width:460px;
  height:320px;
  position:absolute;  /* 绝对定位 */}
```

上面代码定义了qiu标记的显示样式，例如外边距、背景图片、宽度、高度和绝对定位。在IE9.0中，浏览效果如图17-20所示，可以看到页面显示一段文字，无任何格式。

04 添加CSS代码，修饰title和author元素。

```
title{
  font-size:30px;
  color:green;
  position:absolute;
  left:140px;
  top:20px;
}
author{
  font-size:18px;
```

```
    color:#4f2b00;
    position:absolute;
    left:160px;
    top:50px;
  }
```

上面CSS代码定义了title标记的显示样式，例如字体大小、颜色、绝对定位和坐标位置。定义了author标记的显示样式，例如字体大小、字体颜色、绝对定位和坐标位置。

在IE9.0中，浏览效果如图17-21所示，可以看到页面中的标题和作者信息，在指定位置显示。



图17-20　定义背景



图17-21　定义标题和作者信息

05 添加CSS代码，修饰cotent元素。

```
content{
    position:absolute;   /* 绝对定位 */
    font-size:24px;
    line-height:30px;   /* 行间距 */
    left:40px;
    top:70px;
                  font-family:"幼圆";}
br{   display:block;     /* 让诗句分行显示 */}
```

在CSS文件中，定义了content元素的显示样式，例如字体大小、行高、坐标位置等。这里的
标记也是起到了换行作用，当将标记的display属性定义为block块时，可以起到换行的效果。在IE9.0中，浏览效果如图17-18所示，可以看到页面在指定位置显示古诗，并且标题颜色为绿色，作者信息颜色为黄色。



17.7 专家解惑

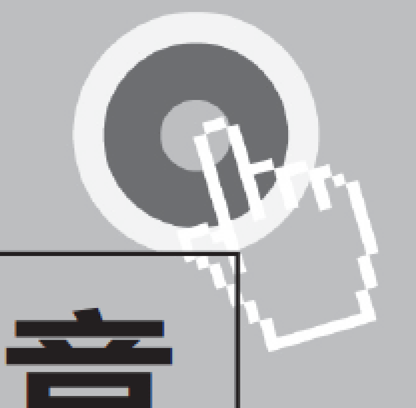
1．背景色和背景图片放在一起定义好吗？

在设定背景图片时，最好同时也设定背景色，这样当背景图片因某种原因无法正常显示时，可以使用背景色来代替。当然，如果正常显示，背景图片会覆盖背景色。

2．XML和HTML文件有什么相同和不同之处？

HTML和XML都是从SGML发展而来的标记语言，因此，它们有些共同点，如相似的语法和标记。不过HTML是在SGML定义下的一个描述性语言，只是一个SGML的应用。而XML是SGML的一个简化版本，是SGML的一个子集。

XML是用来存放数据的，XML不是HTML的替代品，XML和HTML是两种不同用途的语言。XML是被设计用来描述数据的，HTML只是一个显示数据的标记语言。

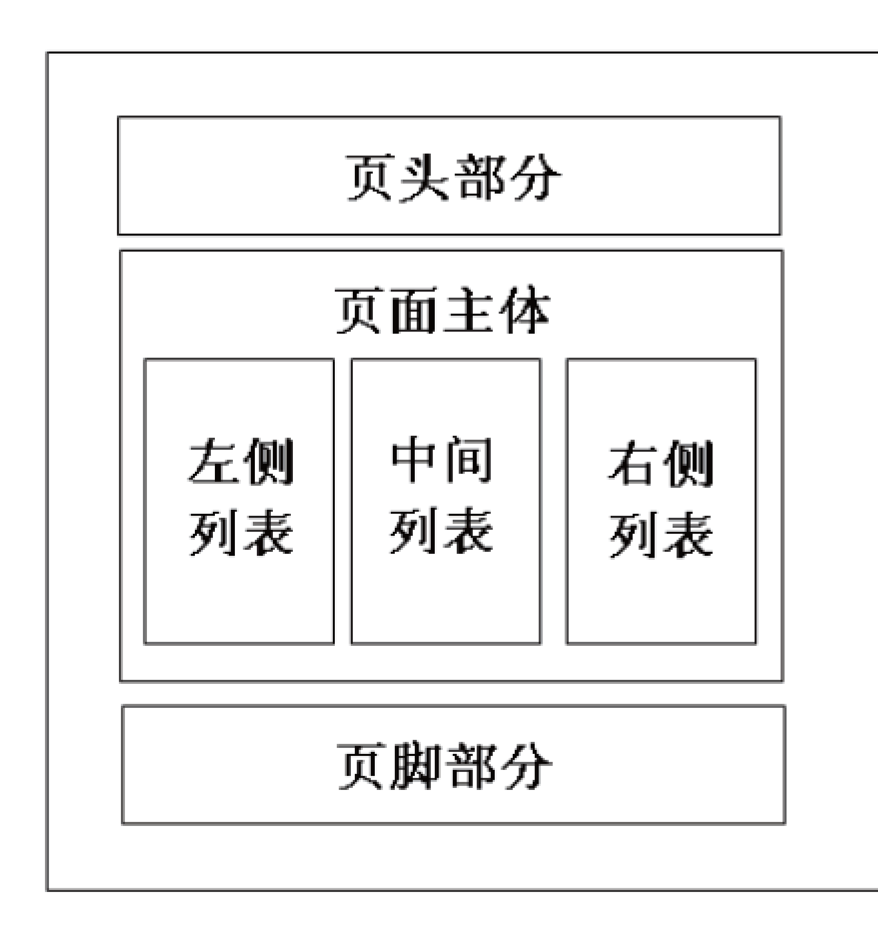


第 18 章

心灵深处的旋律——音乐网站

本章引言

网上听音乐已经成为一种时尚，许多网友为了方便，就直接在网站下载试听。但在国外，如果要听音乐，就需要去指定的音乐商店购买，从商店中挑选自己喜欢的音乐。本章以一个外国音乐网站为例，综合介绍页面布局、DIV排版的制作方法。







18.1 构思布局

本实例采用喜庆、大方的红色为主题，配上多个图片、列表显示各种音乐种类。在IE9.0中浏览效果如图18-1所示。



图18-1 音乐商店首页

18.1.1 设计分析

音乐商店首页是整个网站的第一印象，通常需要设计的大方、合理，能够最大限度地显示页面导航和音乐介绍。一般情况下，首页都是概况性介绍，各个子页面也可以给出链接。设计的重点是布局规范、图文结合漂亮等。

从图18-1所示的页面效果上可以看出，页面总体上划分为上中下结构，上面为页头部分，中间为页面主体内容,下面为页脚。页头部分包括两个部分，分别是页面导航链接和商店介绍。页面主体又分为左中右3个版式，即使用div层将页面主体划分为3个并列区域。页脚部分比较简单，只是一个版权信息。

18.1.2 排版架构

音乐网站比较常见，排版方式也是多种多样，但其本质基本一样，即包含内容比较类似。例如包含导航菜单、音乐列表、音乐新闻、音乐评论和具有自己风格的公司Logo等。本实例也包含了上面这些信息，其版式如图18-2所示。

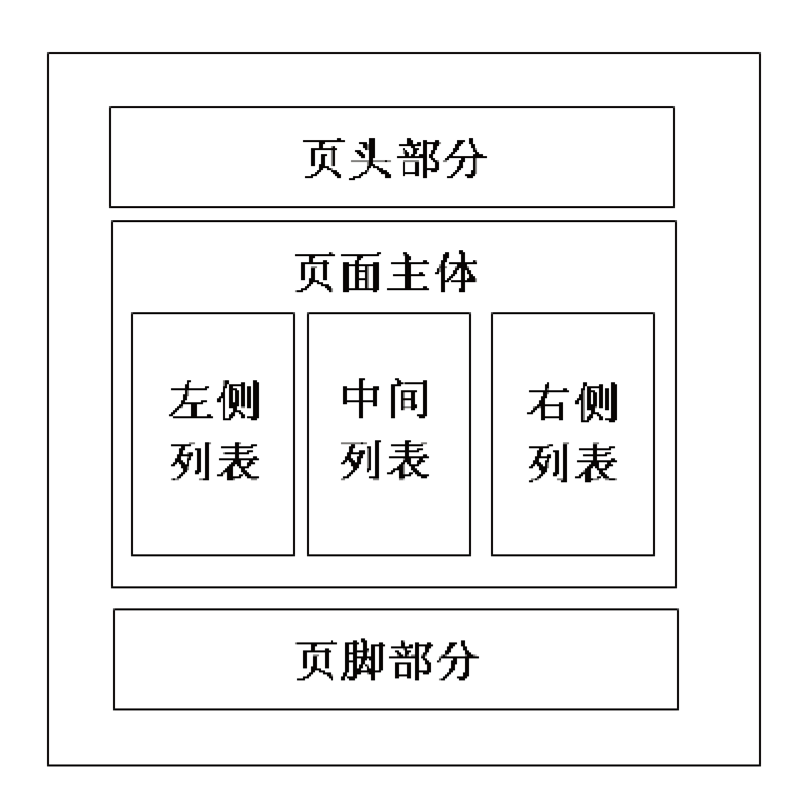


图18-2 页面框架

在本实例中，使用div层进行模块划分，其代码如下：

```
<div id="wrapper">
  <div id="nav">/*导航菜单*/
  </div>
  <div id="topcon">/*网站介绍*/
    </div>
  </div>
  <div id="content">
    <div id="body">/*页面主体*/
      <div class="box" id="news">        /*页面主体左侧*/
      </div>
      <div class="box" id="hits">        /*页面主体中间*/
      </div>
      <div class="box" id="new">         /*页面主体右侧*/
      </div>
     </div>
    <div id="footer">      </div>        /*页脚部分*/
  </div>
</div>
```

上面的各个子块部分，直接对应了HTML代码中的各个div层。#nav和#topcon共同组成页头部分，包含导航菜单、背景图片和网站介绍。#footer是页脚部分，比较简单，这里将其包含在content层里面，#wrapper是整个网页的布局容器。#body中包含了页面主体，里面包含音乐新闻、音乐列表和新发布等列表。

页面主体中是左中右版式，每个版式都采用列表形式显示，如图18-3所示。

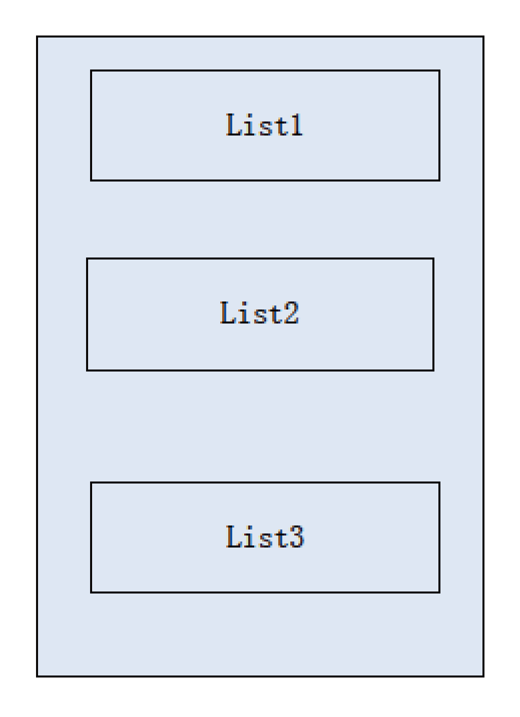


图18-3 列表方式显示

在CSS样式文件中，对上面的div层的修饰代码如下：

```
/** layout **/
#wrapper {
    width: 678px;
    min-height: 750px;
    _height: 750px;
    background: url(images/header.jpg) no-repeat;
  position: relative;
}

h1 {
  padding: 25px 0 0 30px;
  font: 32px "arial black", arial, sans-serif;
  color: #151515;
}
h1 em {
  color: #ffffff;
  font-weight: bold;
  font-style: normal;
  position: relative;
  top: -4px;
}
#content {
  width: 710px;
  position: absolute;
  color: #fff;
  top: 299px;
  left: 33px;
}
#content a {
  color: #fff;
}
#content a:hover {
  color: #fee;
}
```

在上面代码中，#wrapper选择器定义了整个布局容器显示，例如宽度、背景图片、对

齐方式，并使用min-height属性设置层的最小高度为750px，但IE浏览器不支持这个属性。#content选择器中定义了宽度、对齐方式、字体颜色和坐标位置。



18.2 模块分割

页面框架整体布局完成之后，就可以对各个模块进行分别处理，最后再统一整合、调整样式。这也是设计、制作网站的一般步骤。

18.2.1 页头部分

本实例页头部分包含两个方面的内容，一个是导航菜单；另一个是音乐商店介绍。其中，商店介绍部分包含了文字和图片。页头部分效果如图18-4所示。



图18-4 页头部分

创建HTML页面，其中实现页头部分的HTML代码如下：

```
        <h1><em><span lang="zh-cn">音乐</span></em> <span lang="zh-cn">商店</
span></h1>
        <div id="nav">
          <ul>
            <li><a href=""><span class="style2">01</span> </a>
             <span lang="zh-cn" class="style1"><a href="index.html">关于商店</
a></span></li>
            <li><a href=""><span>02</span> </a>
             <span lang="zh-cn" class="style1"><a href="index.html">画廊</a></
span></li>
            <li><a href=""><span>03</span> </a>
             <span lang="zh-cn" class="style1"><a href="index.html">免费音乐</
a></span></li>
            <li style="height: 39px"><a href=""><span>04</span> </a>
             <span lang="zh-cn" class="style1"><a href="index.html">艺术家</
```

```
a></span></li>
            <li><a href=""><span>05</span> </a>
             <span lang="zh-cn" class="style1"><a href="index.html">联系我们</
a></span></li>
          </ul>
        </div>
          <div id="topcon">
          <div id="topcon-inner">
            <h2><span lang="zh-cn">欢迎</span> <span lang="zh-cn">到音乐商店</
span></h2>
             <p>
                    <span id="result_box" lang="zh-CN" class="short_text"
closure_uid_h11zq="136" c="4" a="undefined" kd="null">
                    <span closure_uid_h11zq="112" kd="null">在这里可以找到自己喜欢
的音乐</span><span closure_uid_h11zq="116" kd="null"></span></span></p>
              <p>
             <span id="result_box0" lang="zh-CN" class="short_text" closure_
uid_h11zq="136" c="4" a="undefined" kd="null">
             <span closure_uid_h11zq="189">可以</span><span closure_uid_
h11zq="190">从</span><span closure_uid_h11zq="191">这个模板</span><span closure_
uid_h11zq="192">中</span><span closure_uid_h11zq="193">删除</span><span
closure_uid_h11zq="194">我们网站</span><span closure_uid_h11zq="195">的</
span><span closure_uid_h11zq="196">任何</span><span closure_uid_h11zq="197">链
接，可以免费使用这些模板。</span></span></p>
          </div>
        </div>
```

上面代码中，首页使用标题h1，定义本网站标识，即“音乐商店”。在ID名称为nav的层中，使用无序列表创建了导航菜单，用于链接网站中其他的子页面。在ID名称为topcon的层中，使用文本信息介绍了本商店的主体内容。

添加CSS代码，定义页头显示样式，代码如下：

```
    #nav {
      position: absolute;
      top: 0px;
      left: 335px;
      width: 500px;
    }
    #nav li {
      float: left;
      background: url(images/nav_left.gif) no-repeat;
      list-style: none;
      padding-left: 10px;
      padding-right: 20px;
      padding-top: 45px;
      line-height: 1.1;
    }
    #nav span {
      display: block;
      font-size: 11px;
    }
```

```
#nav a {
  color: #FFFFFF;
  font-size: 11px;
  font-weight: bold;
  text-decoration: none;
}
/** topcontent **/
#topcon {
  background: url(images/topcon.jpg) no-repeat;
  width: 427px;
  position: absolute;
  top: 105px;
  left: 338px;
  color: #fff;
}
#topcon-inner {
  margin: 33px 40px 41px 85px;
  height: 120px;
  overflow: auto;
}
#topcon h2 {
  font-size: 14px;
}
```

#nav样式中定义了层nav的整体样式，例如宽度、绝对定位和坐标位置。#nav li样式中定义了列表选项的显示样式，例如行高、浮动定位，背景图片、列表特殊符号和内边距等。#nav span定义了span元素以块显示，字体大小为11px。#nav a定义了超级链接显示样式，例如字体颜色、字体大小、下划线和字体样式等。#topcon定义了页头部分的背景图片、宽度、相对定位、坐标位置和字体颜色等。#topcon-inner定义了外边距和高度。#topcon h2定义了字体大小。

18.2.2 左侧内容列表

页面主体左侧显示的是音乐新闻，例如当前的音乐盛会和音乐专辑出版等。可以包含文本和图片信息等。在IE9.0中浏览效果如图18-5所示。



图18-5 页面左侧列表

在HTML网页中，实现上面效果的HTML代码如下：

```
        <div id="content">
          <div id="body">
            <div class="box" id="news">
              <div class="box-t"><div class="box-r"><div class="box-
    b"><div class="box-l">
              <div class="box-tr"><div class="box-br"><div class="box-
    bl"><div class="box-tl">
                <h2><span lang="zh-cn">新闻</span> &amp; <span lang="zh-cn">事
件</span></h2>
                <h3>06.03.2011</h3>
                 <p><span lang="zh-cn">李尔·韦恩新专辑曲目曝光 将于8月29日上市。</
span></p>
                <p class="more"><span lang="zh-cn">更多</span><a href=" ">...</
a></p>
                <div class="hr-yellow"> </div>
                <h3>06.03.2011</h3>
                <p>
                    <span id="result_box1" lang="zh-CN" class="short_text"
closure_uid_h11zq="136" c="4" a="undefined" kd="null">
                    <span closure_uid_h11zq="307" kd="null">大运村里欢乐多：体验
中国传统文化 美妙音乐减压</span><span closure_uid_h11zq="314" kd="null"></span></
span></p>
                    <p><span lang="zh-cn"></span>文化体验区有中国传统文化展、油画艺
术展、雕塑艺术展以及汉语学习中心等，外国运动员在这里可以欣赏到中国的传统文化，还可以亲自上阵
体验一把。</p>
                <p class="more"><span lang="zh-cn">更多</span><a href="">...</
a></p>
              </div></div></div></div>
              </div></div></div></div>
            </div>
    </div></div>
```

上面代码中，content层是页面内容的布局容器，这里包含了页面主体和页脚，其样式在后面介绍。body层是页面主体的布局容器，包含了左中右版式布局，上面代码只是列出了左侧列表。下面两个小节介绍的div层，都包含在body层中。

ID名称为news的层，是页面左侧列表的布局容器，其内容都是在此处显示。news层所包含的层，用来定义边框显示样式。

在样式文件中，对于上面层的CSS样式定义代码如下：

```
    .box {
      float: left;
      width: 195px;
      background: #730F11;
      margin-right: 18px;
    }
    .box-t { background: top url(images/box_t.gif) repeat-x; }
    .box-r { background: right url(images/box_r.gif) repeat-y; }
    .box-b { background: bottom url(images/box_b.gif) repeat-x; }
```

```
.box-l { background: left url(images/box_l.gif) repeat-y; }
.box-tr { background: top right url(images/box_tr.gif) no-repeat; }
.box-br { background: bottom right url(images/box_br.gif) no-repeat; }
.box-bl { background: bottom left url(images/box_bl.gif) no-repeat; }
.box-tl { background: top left url(images/box_tl.gif) no-repeat; }
.box-tl {
  padding: 13px 18px;
}
.box p {
  margin: 1em 0;
}
p.more {
  margin: 0;
}
```

上面代码中，类选择器box定义了页面右浮动显示。类选择器box-t定义上面的背景图片，其他依次类推。嵌套选择器box p定义了外边距。

18.2.3 中间内容列表

在页面主体内容中，中间列表包含了音乐网站中，大力推荐的音乐曲目清单，其中包含了文本信息和图片。在此列表中，浏览者可以根据需要选择自己喜欢的曲目，并进入相应的子页面。在IE9.0中浏览效果如图18-6所示。
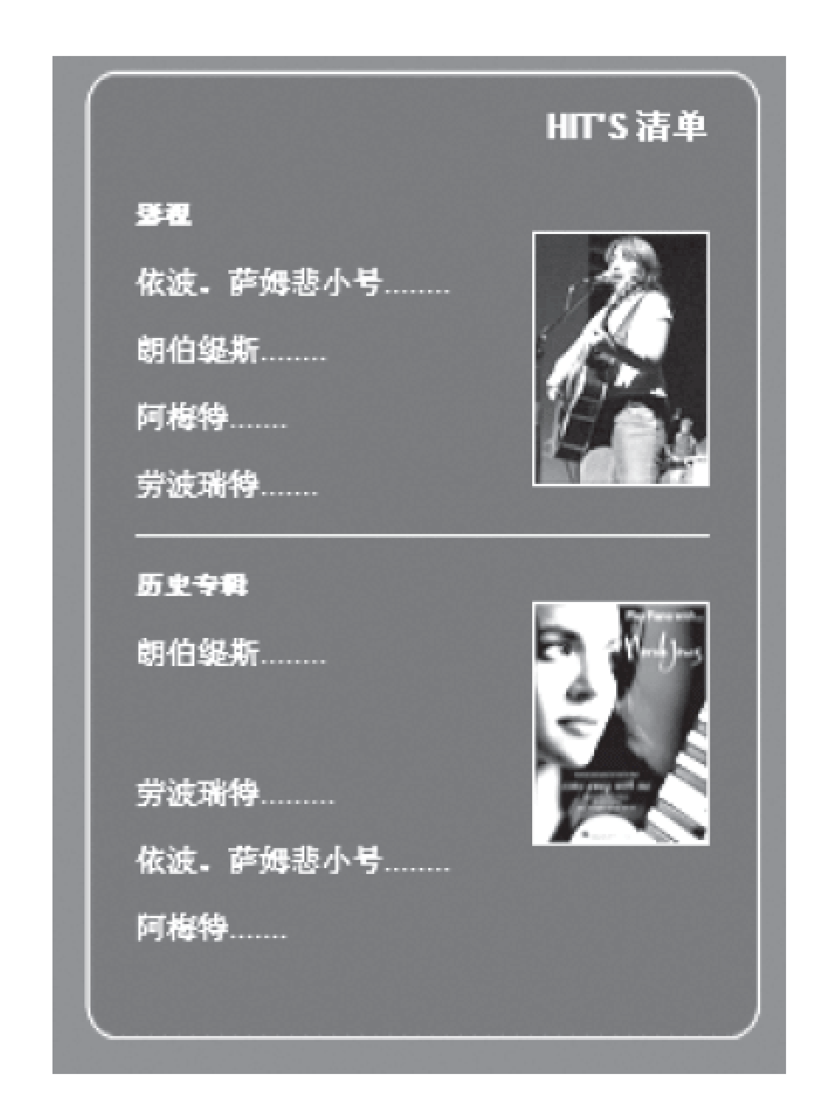


图18-6　页面中间列表

在HTML文件中，实现页面主体中间列表的代码如下：

```
<div class="box" id="hits">
          <div class="box-t"><div class="box-r"><div class="box-
b"><div class="box-l">
          <div class="box-tr"><div class="box-br"><div class="box-
bl"><div class="box-tl">
            <h2>HIT&#39;S <span lang="zh-cn">清单</span></h2>
            <h3><span lang="zh-cn">影视</span></h3>
            <img src="images/pic_1.jpg" width="63" height="91" alt="Pic 1"
```

```
class="right" />
                <ul>
                  <li><span lang="zh-cn">依波。萨姆悲小号</span><a href="
">...</a></li>
                  <li><span lang="zh-cn">朗伯缇斯</span><a href=" ">... </
a></li>
                  <li>
                          <span id="result_box2" lang="zh-CN" class="short_
text" closure_uid_h11zq="136" c="4"
    a="undefined" kd="null">
                          <span closure_uid_h11zq="615" kd="null">阿梅特</
span></span><a href=" ">.......</a></li>
                  <li><span lang="zh-cn">劳波瑞特</span><a href=" ">... </
a></li>
                </ul>
                <div class="hr-yellow"> </div>
                <h3><span lang="zh-cn">历史专辑</span></h3>
                <img src="images/pic_2.jpg" width="63" height="87" alt="Pic 2"
class="right" /><ul>
                  <li style="width: 123px"><span lang="zh-cn">朗伯缇斯</span><a
href=" ">... </a></li>
                </ul>
                 <ul>
                  <li><span lang="zh-cn">劳波瑞特</span><a href=" ">...
</a></li>
                  <li><span lang="zh-cn">依波。萨姆悲小号</span><a href="
">...</a></li>
                  <li>
    <span id="result_box3" lang="zh-CN" class="short_text" closure_uid_
h11zq="136" c="4" a="undefined" kd="null">
       <span closure_uid_h11zq="615" kd="null">阿梅特</span></span><a href="
">...</a></li>
                </ul>
              </div></div></div></div>
              </div></div></div></div>
            </div>
```

上面代码中，hits层实际是包含在body层中的，用于显示页面主体的中间内容。在hits层中，同样使用box类选择器定义中间列表的显示样式。hits层包含了图片和无序列表信息。

在样式文件中，对于上面层的CSS样式定义如下：

```
    #hits {
      width: 240px;
    }
    #hits .box-tl {
      padding-bottom: 22px;
    }
    #hits ul {
      margin: 1em 0;
    }
    #hits li {
```

```
      list-style: none;
      margin: 0.9em 0;
    }
    #hits ul a {
      text-decoration: none;
    }
    #hits ul a:hover {
      text-decoration: underline;
    }
```

上面代码中，#hits选择器定义了中间部分的宽度为240像素，#hits .box-tl和#hits ul选择器分别定义了内边距和外边距。#hits li选择器定义了列表选项的显示样式，例如无特殊符号显示和外边距等。#hits ul a定义了列表超级链接显示样式，即不带下划线。#hits ul a:hover选择器定义了列表中超级链接的悬浮样式，即鼠标放到链接上时，显示下划线等。

18.2.4 右侧内容列表

在音乐网站首页中，右侧内容主要显示音乐专辑发布信息，例如某某发布最新专辑信息。右侧内容可以包含文本信息和图片信息，通过相应链接可以进入新专辑页面信息。右侧内容列表，在IE9.0中，浏览效果如图18-7所示。



图18-7　右侧内容列表

在HTML页面中，实现右侧内容的HTML代码如下：

```
      <div class="box" id="new">
             <div class="box-t"><div class="box-r"><div class="box-
    b"><div class="box-l">
             <div class="box-tr"><div class="box-br"><div class="box-
    bl"><div class="box-tl">
               <h2> <span lang="zh-cn"> 新 发布</span></h2>

               <h3><span lang="zh-cn">伊塔撒德</span></h3>
               <img src="images/pic_3.jpg" width="66" height="52" alt="Pic 3"
class="right" />
               <p><span lang="zh-cn">波特扭黎.....</span>. </p>
                   <p>
```

```
                    <span id="result_box4" lang="zh-CN" class="short_text"
closure_uid_h11zq="136" c="4" a="undefined" kd="null">
                    <span closure_uid_h11zq="681">梅塞纳斯 </span></span>quam.
Sed</p>
                <h3><span lang="zh-cn">曼撒纳斯</span></h3>
                <img src="images/pic_4.jpg" width="66" height="52" alt="Pic 4"
class="right" />
                <p><span lang="zh-cn">纳迪斯</span> sollicitudin 
                    <span lang="zh-cn">咔哇里斯</span>convallis</p>
                <h3><span lang="zh-cn">撒黎缇斯</span></h3>
                <img src="images/pic_5.jpg" width="66" height="52" alt="Pic 5"
class="right" />
                <p><span lang="zh-cn">桑达.纳斯</span> nvallis <span lang="zh-
cn">劳特.可瓦斯</span> .vallislacus
                    <span lang="zh-cn">阿里瓦斯</span>.vallis</p>
              </div></div></div></div>
              </div></div></div></div>
            </div>
            <div class="clear"> </div>
```

在上面代码中，new层是页面右侧内容的布局容器，其文本和图片信息都在此层中显示。在页面最后一行中，创建了一个clear层。

在样式文件中，对于上面层的CSS样式定义如下：

```
#new {
  margin-right: 0;
}
#new .box-tl {
  padding-bottom: 18px;
}
#new p {
  margin-top: 0;
  margin-bottom: 3.6em;
}
.clear {
    clear: both;
}
```

上面代码中，#new选择器定义了new层的右外边距距离，下面的#new .box-tl和#new p分别定义了底部内边距和上下外边距距离。使用类选择器clear，用于消除float浮动布局带来的影响。

18.2.5 页脚部分

本实例的页脚部分非常简单，但作为一个必不可少的元素，又不得不介绍。页脚部分主要显示版权信息和地址信息，在IE9.0中，浏览效果如图18-8所示。

©联系我们

图18-8 页脚部分

在HTML文件中，实现页脚部分的HTML代码如下：

```
<div id="footer">
<p>© <span lang="zh-cn"><a href="index.html">联系我们</a></span></p>
</div>
```

在CSS样式文件中，定义footer部分的CSS代码如下：

```
#footer {
  text-align: center;
}
#footer p {
  margin: 0.8em;
}
```

上面代码中，#footer定义了页脚对齐方式，#footer p选择器定义了外边距距离。



18.3　整体调整

通过上面对各个模块的定义，各个模块的基本样式已经具备，页面基本成形。在制作完成的最后，还需要对页面效果进行一些细微的调整，例如各块之间的padding和margin值是否与整体页面协调，各个子块之间是否统一等。

18.3.1　页面内容主体调整

虽然前面使用CSS定义了页面内容样式，即左侧、中间和右侧内容列表的显示样式，但其整体样式并没有进行定义，例如对body层的样式修饰等。在没有使用CSS代码对样式进行定义之前，在IE9.0中，浏览效果如图18-9所示。



图18-9　body样式定义前

在CSS样式文件中，添加代码定义body主体内容的显示样式，其代码如下：

```
#body {
  border: 3px solid white;
  background: #901315;
  padding: 18px;
}
#body h2 {
  font-size: 12px;
```

```
    text-align: right;
    margin-bottom: 1.5em;
  }
  #body h3 {
    font-size: 9px;
    color: #FFEA00;
  }
  #body .more a {
    font-weight: bold;
    text-decoration: none;
  }
  #body .more a:hover {
    text-decoration: underline;
  }
  #body .hr-yellow {
    border-top: 1px solid #FFEA00;
    padding-bottom: 1em;
    margin-top: 1em;
  }
```

在上面的代码中，#body选择器定义了边框样式、背景色和外边距距离。#body h2定义了标题h2的显示样式，例如字体大小、对齐方式和底部外边距距离。#body h3选择器定义了标题h3的显示样式，例如字体大小和字体颜色。#body .more a选择器定义了字体是否加粗和带有下划线。

CSS样式代码添加后，在IE9.0中浏览效果如图18-10所示。可以发现文字标题发生变化，都变为黄色，每个选项之间都用浅黄色边框隔开，并且文字变小。

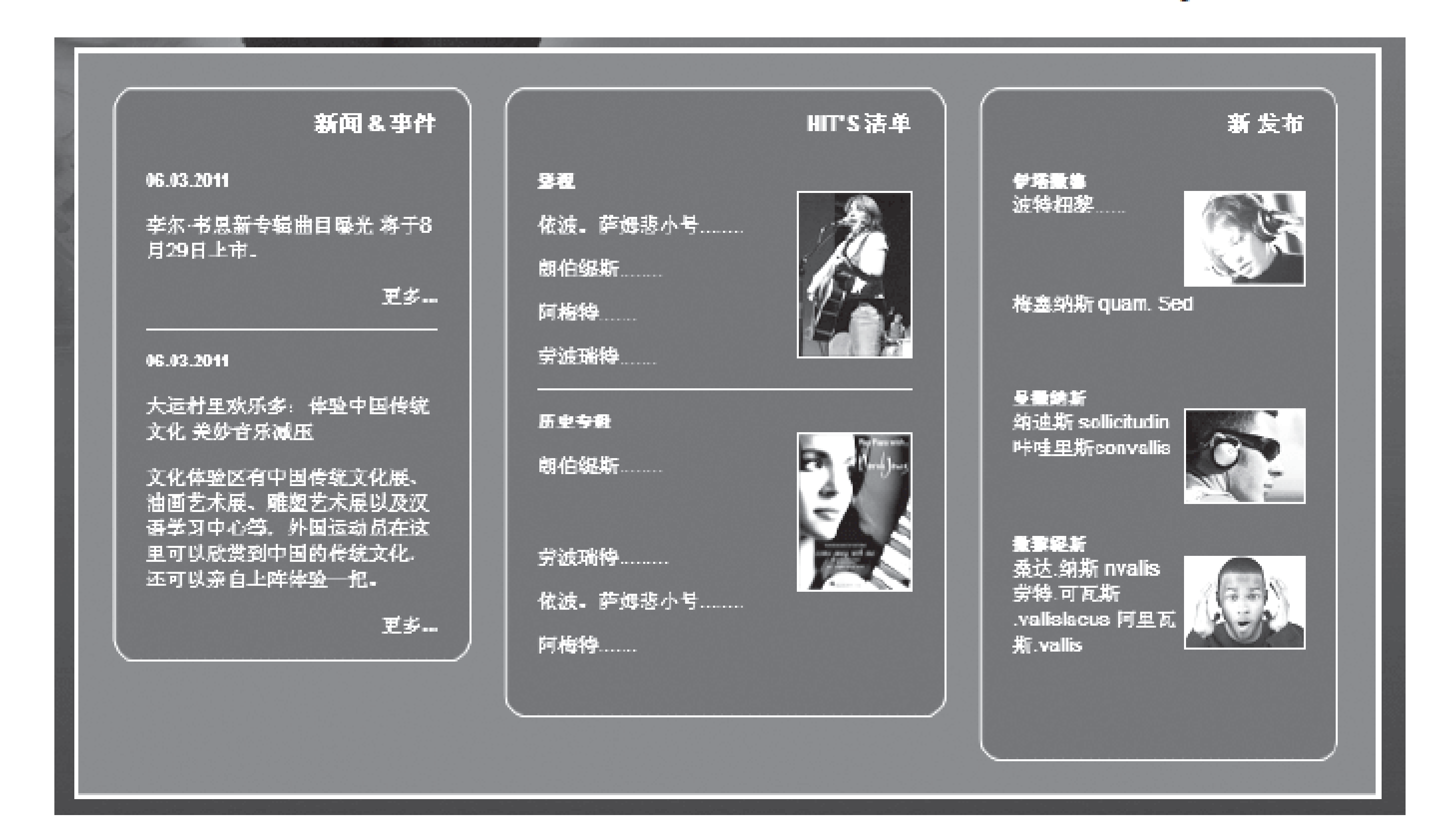


图18-10 body样式定义后

18.3.2 页面整体调整

最后，我们就可以对页面整体样式进行统一和协调了。例如设置全局文本样式、对齐方式和内外边距等。还可以对页面中的内容块进行大小调整。

在CSS样式文件中，其代码如下：

```
  html, body, h1, h2, h3, h4, ul, li {
     margin: 0;
     padding: 0;
  }
  h1 img {
     display: block;
```

```
}
img {
    border: 0;
}
a {
    color: #FFFFFF;
}
a:hover {
    color: #FFA405;
}
.left {
    float: left;
}
.right {
    float: right;
}
.more {
    text-align: right;
}
body {
    background: #3A0404 url(images/page_bg.jpg) repeat-x;
    font: 11px arial, sans-serif;
    color: #464544;
    padding-bottom: 10px;
}
```

在上面代码的body标记选择器中，定义了背景图片、背景颜色、字体样式和底部内边距距离等。其他标记选择器比较简单，这里就不再介绍。

整体样式设置完成后，页面在IE9.0中浏览效果如图18-1所示。



18.4 专家解惑

1．有时候显示一个三层并列网页，在IE和Firefox浏览器的显示效果为什么不同？

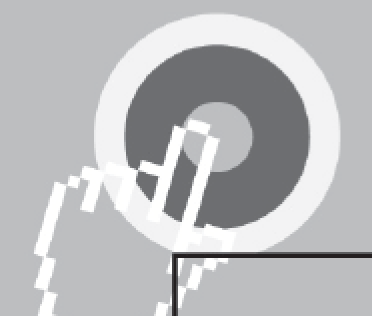
很多时候，尤其是容器内有平行布局，例如有两、三个float的div时，宽度很容易出现问题。在IE浏览器中，外层的宽度会被内层更宽的div挤破。此时，一定要用Photoshop或者Firework量取像素级的精度。

2．在IE浏览器中，图片和下方父元素产生间隙该怎么解决？

在IE浏览器中，如果父元素直接包含<img>，这张图片下方会和父元素边缘产生间隙。

其解决办法有两个，一是在源代码中让</div>和<img>在同一行，因为那个间隙是由换行符产生的。二是给<img>添加样式，其代码为

```
display:block;
```



第 19 章

企业的门面——小型企业网站开发

本章引言

作为小型软件企业的网站，一般规模不是太大，通常包含3~5个栏目，例如产品、客户和联系我们栏目等，并且有的栏目甚至只包含一个页面，例如联系我们栏目。此类网站通常都是为了展示公司形象，说明一下公司的业务范围和产品特色等，一般实现这样的网站就是一个首页加上若干内容即可。





19.1 构思布局

本实例是模拟一个小型软件公司的网站，其公司主要承接电信方面的各种软件项目。网站上包括首页、产品信息、客户信息和联系我们等栏目。本实例中采用红色和白色配合使用，红色部分显示导航菜单，白色显示文本信息。在IE9.0中，浏览效果如图19-1所示。



图19-1 计算机网站首页

19.1.1 设计分析

作为一个软件公司网站首页，其页面需要简单、明了，给人以清晰的感觉。页头部分主要放置导航菜单和公司Logo信息等，其Logo可以是一张图片或者文本信息等；页面主体分为两个部分，页面主体左侧是公司介绍，对于公司介绍可以在首页上概括性描述；页面主体右侧是新闻、产品和客户信息等，其中，产品和客户的链接信息，以列表形式对重要信息进行介绍，也可以通过页面顶部导航菜单进入相应页面介绍。

对于网站的其他子页面，篇幅可以比较短，其重点是介绍软件公司业务、联系方式、产品信息等，页面与首页风格相同即可。

19.1.2 排版架构

从图19-1可以看出，页面结构并不是太复杂，采用的是上中下结构，页面主体部分又嵌套了一个左右版式结构。其效果如图19-2所示。

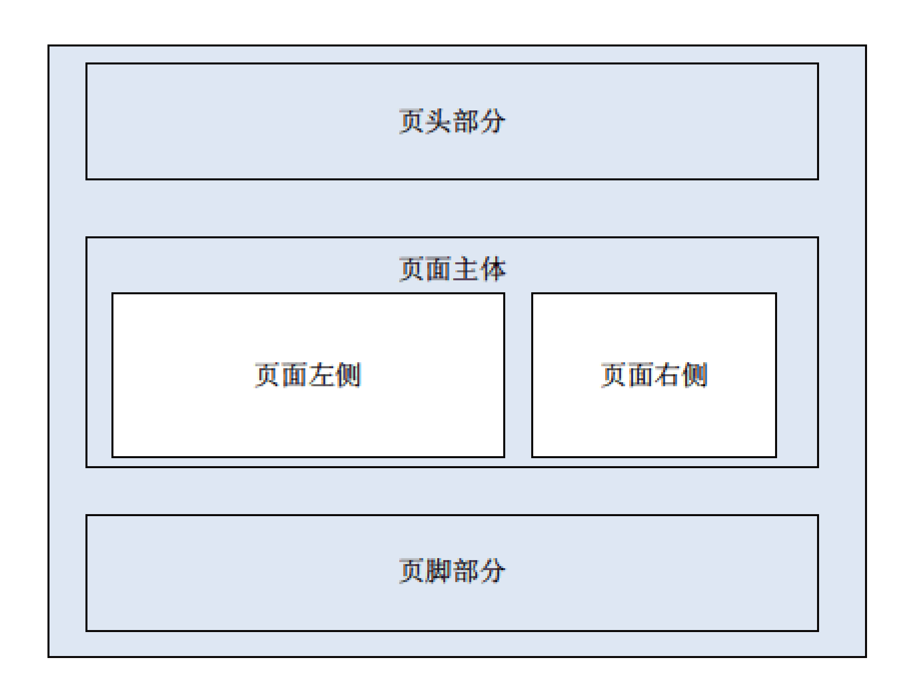


图19-2 页面总体框架

在HTML页面中，通常使用div层对应上面不同的区域，可以是一个div层对应一个区域，也可以是多个div层对应同一个区域。本实例的div代码如下所示。

```
<div id="container">/*页面布局容器*/
<div id="top">
</div><!--end top-->
<div id="header">
</div><!--end header-->
<div id=me>             /*导航菜单*/
</div>
<div id="content">
<div id="text">         /*页面主体左侧内容*/
</div><!--end text-->
<div id="column">       /*页面主体右侧内容*/
</div><!--end column-->
</div><!--end content-->
<div id="footer">       /*页脚部分*/
</div><!--end footer-->
</div><!--end container-->
```

上面代码中，ID名称为container的层是整个页面的布局容器，top层、header层和me层共同组成了页头部分，top层用于显示页面Logo，header层用于显示页头文本信息，me层用于显示页头导航菜单信息。页面主体是content层，其包含了两个层：text层和column层。text层是页面主体左侧内容，显示公司介绍信息；column层是页面主体右侧内容，显示公司常用的导航链接。footer层是页脚部分，用于显示版权信息和地址信息。

在CSS文件中，对应container和content层的CSS代码如下：

```
#container
{
 margin: 0pt auto;
width: 770px;
 position: relative; }
#content {
 background: transparent url('images/content.gif') repeat-y;
 clear: both;
 margin-top: 5px;
width: 770px;
}
```

上面代码中，#container选择器定义了整个布局容器的宽度、外边距和定位方式。#content选择器定义了背景图片、宽度和顶部边距。



19.2 模块分割

当页面整体架构完成后，就可以动手制作不同的模块区域了。其制作流程，采用自上而下，从左到右的顺序。完成后，再对页面样式进行整体调整。

19.2.1 Logo与导航菜单

一般情况下，Logo信息和导航菜单都是放在页面顶部，作为页头部分。其中Logo信息作为公司标志，通常放在页面的左上角或右上角；导航菜单放在页头部分和页面主体二者之间，用于链接其他的页面。在IE9.0中浏览效果如图19-3所示。



图19-3　页面Logo和导航菜单

在HTML文件中，用于实现页头部分的HTML代码如下：

```
<div id="top">
</div><!--end top-->
<div id="header">
<h1>计算机 网站</h1>
</div><!--end header-->
<div id=me>
<ul id="menu">
<li><a href="#" class="actual">首页</a></li>
<li><a href="#" >产品</a></li>
<li><a href="#">客户</a></li>
<li><a href="#">联系方式</a></li>
</ul>
</div>
```

上面代码中，top层用于显示页面Logo；header层用于显示页头的文本信息，例如公司名称；me层用于显示页头导航菜单。在me层中，有一个无序列表，用于制作导航菜单，每个选项都是由超级链接组成的。

在CSS样式文件中，对应上面标记的CSS代码如下：

```
#top {
background: transparent url('images/top.jpg') no-repeat;
height: 50px;
}
#top p {
 margin: 0pt;
padding: 0pt;
}
#header {
 background: transparent url('images/header.jpg') no-repeat;
 height: 150px;
 margin-top: 5px;
}
#menu {
 position: absolute;
top: 180px;
left: 15px;
}
#header h1 {
 margin: 5px 0pt 0pt 50px;
padding: 0pt;
font-size: 1.7em;
 }
#header h2 {
margin: 10px 0pt 0pt 90px;
 padding: 0pt;
 font-size: 1.2em;
color: rgb(223, 139, 139);
}
ul#menu {
margin: 0pt;
}
#menu li {
list-style-type: none;
float: left;
text-align: center;
 width: 104px;
margin-right: 3px;
 font-size: 1.05em;
}
#menu a {
 background: transparent url('images/menu.gif') no-repeat;
overflow: hidden;
 display: block;
height: 28px;
padding-top: 3px;
 text-decoration: none;
 twidth: 100%;
font-size: 1em;
font-family: Verdana,"Geneva CE",lucida,sans-serif;
color: rgb(255, 255, 255);
```

```
    }
    #menu li > a, #menu li > strong {
     width: auto;
     }
    #menu a.actual {
    background: transparent url('images/menu-actual.gif') no-repeat;
    color: rgb(149, 32, 32);
    }
    #menu a:hover {
     color: rgb(149, 32, 32);
     }
```

上面代码中，#top选择器定义了背景图片和高度；#header定义了背景图片、高度和顶部外边距；#menu层定义了层定位方式和坐标位置。其他选择器分别定义了上面三个层中元素的显示样式，例如段落显示样式、标题显示样式、超级链接样式等。

19.2.2 左侧文本介绍

在页面主体中，其左侧版式主要介绍公司相关信息。左侧文本采用的是左浮动并且固定宽度的版式设计，重点在于调节宽度使不同浏览器之间能够效果一致，并且颜色上配合Logo和左侧的导航菜单，使整个网站和谐、大气。在IE9.0中浏览效果如图19-4所示。



图19-4 页面左侧文本介绍

在HTML文件中，创建页面左侧内容介绍的代码如下：

```
    <div id="content">
    <div id="text">
    <h3 class="headlines"><a href="#" title="testing">欢迎来到我们的网站 </
a></h3>
    <p><img src="images/fotos.jpg" alt="fotos" align="right" />
    远大公司成立于1998年，注册资本1700万元。是国家认定的高新技术企业、软件企业，是专业的电
信系统仿软件和应用服务供应商。</p><p>
    公司坚持走自立创新、稳步发展的道路，以创立品牌为自己的基本策略，以产品自身的品质，先进的
技术和良好的服务取信于用户。2002年至今公司先后有多个软件产品获得了河南省信息产业厅颁发的《软
件产品登记证书》和国家版权局颁发的《软件著作权登记证书》。同时远大的进步和发展，也得到了政府
部门的大力支持和关注，获得国家科技部和省、市政府部门技术创新基金无偿资助百余万元。并正式获得
中国质量体系认证中心颁发的ISO9001:2008质量管理体系认证证书。
```

```
</p>
<p> </p>
</div><!--end text-->
</div>
```

上面代码中，content层是页面主体，text层是页面主体中的左侧部分。text层包含了标题和段落信息，段落中包含一张图片。

在CSS文件中，对于上面HTML标记的CSS代码如下：

```
#text {
background: rgb(255, 255, 255) url('text-top.gif') no-repeat;
 width: 518px;
color: rgb(0, 0, 0);
float: left;
}
#text h1, #text h2, #text h3, #text h4 {
 color: rgb(140, 9, 9);
}
#text h3.headlines a {
 color: rgb(140, 9, 9);
}
```

上面代码中，#text层定义了背景图片、背景颜色、字体颜色和页面左浮动。#text下面的选择器定义了标题显示样式，例如字体颜色等。#text h3.headlines a选择器定义了标题3、类headlines和超级链接显示样式。

19.2.3 右侧导航链接

在页面主体右侧版式中，文本信息不是太多，但非常重要。它是首页用于链接其他页面的导航链接，例如客户详细信息、最新消息等。同样右侧版式需要设置为固定宽度并且向左浮动。在IE9.0中浏览页面效果如图19-5所示。

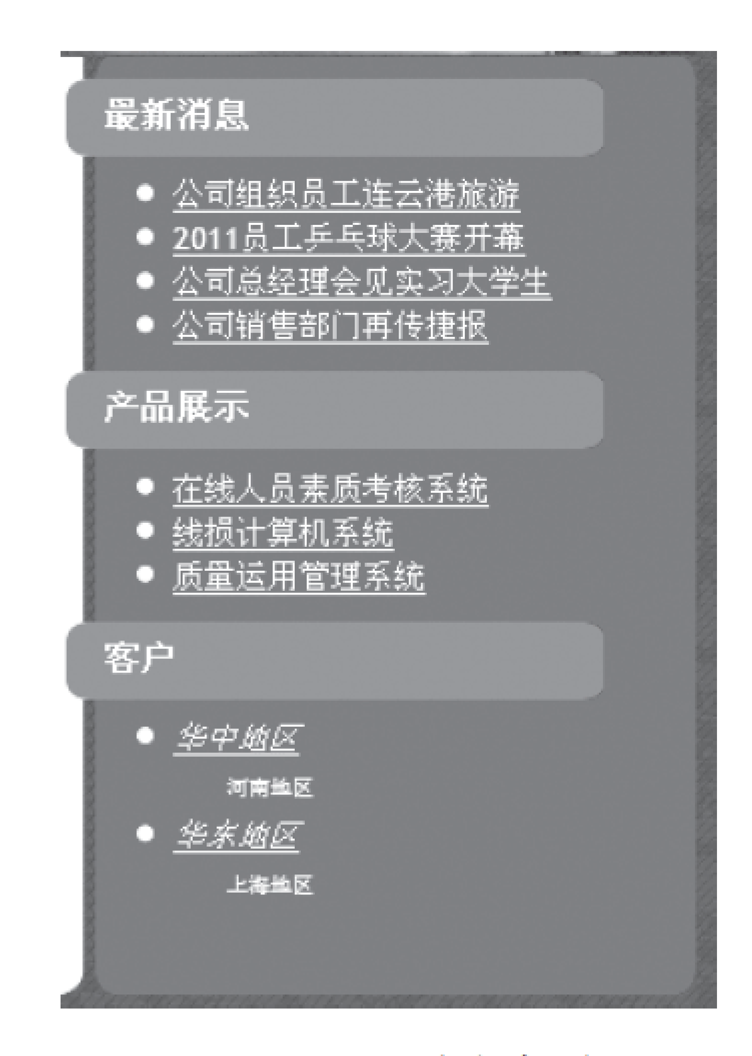


图19-5 页面右侧链接

从图19-5可以看出，包含几个无序列表和标题，其中列表选项为超级链接。HTML文件中用于创建页面主体右侧版式的代码如下：

```
    <div id="column">
    <h3><span>最新消息</span></h3>
    <ul class="category_list"><li><a href="#">公司组织员工连云港旅游</a></li>
    <li><a href="#">2011员工乒乓球大赛开幕</a></li>
    <li><a href="#">公司总经理会见实习大学生</a></li>
    <li><a href="#">公司销售部门再传捷报</a></li></ul>
    <h3><span>产品展示</span></h3>
    <ul class="recent_articles"><li><a href="#">在线人员素质考核系统</a></li>
    <li><a href="#">线损计算机系统</a></li>
    <li><a href="#">质量运用管理系统</a></li></ul>
    <h3><span>客户</span></h3>
    <ul class="wet_recent_comments"><li><a href="#"><cite>华中地区</cite></
a><p>河南地区</p></li>
    <li><a href="#"><cite>华东地区</cite></a><p>上海地区</p></li></ul>
    </div><!--end column-->
    <div id="content-bottom"> </div>
```

在上面的代码中，创建了两个层，分别为column层和content-bottom层。其中column层用于显示页面主体中的右侧链接，并包含了3个标题和3个超级链接。content-bottom层用于消除上面层的float浮动效果。

在CSS文件中，用于修饰上面HTML标记的CSS代码如下：

```
    #column {
     background: rgb(142, 14, 14) url('images/column.gif') no-repeat;
    float: right;
     width: 247px; }
    #column p { font-size: 0.7em; }
    #column ul { font-size: 0.8em; }
    #column h3 {
    background: transparent url('images/h3-column.gif') no-repeat;
    position: relative;
    left: -18px;
     height: 26px;
    width: 215px;
    margin-top: 10px;
     padding-top: 6px;
    padding-left: 6px;
    font-size: 0.9em;
    z-index: 1;
    font-family: Verdana,"Geneva CE",lucida,sans-serif;
    }
    #column h3 span { margin-left: 10px; }
    #column span.name {
     text-align: right;
     color: rgb(223, 58, 0);
    margin-right: 5px;
    }
```

```
    #column a { color: rgb(255, 255, 255); }
    #column a:hover { color: rgb(80, 210, 122); }
    p.comments {
    text-align: right;
    font-size: 0.8em;
     font-weight: bold;
     padding-right: 10px;
    }
    #content-bottom {
    background: transparent url('images/content-bottom.gif') no-repeat scroll
left bottom;
    clear: both;
     display: block;
    width: 770px;
    height: 13px;
     font-size: 0pt;
    }
```

上面代码中，#column选择器定义了背景图片、背景颜色、页面右浮动和宽度。#content-bottom选择器定义了背景图片、宽度、高度、字体大小和以块显示，并且使用clear消除前面层使用float的影响。其他选择器主要定义column层中其他元素的显示样式，例如无序列表样式、列表选项样式和超级链接样式等。

19.2.4 版权信息

版权信息一般放置到页面底部，用于介绍页面的作者、地址信息等，是页脚的一部分。页脚部分和其他网页部分一样，需要保持简单、清晰的风格。在IE9.0中浏览效果如图19-6所示。

图19-6 页脚部分

从图19-6可以看出，此页脚部分非常简单，只包含了一个作者信息的超级链接，因此设置起来比较方便，其代码如下：

```
    <div id="footer">
    <p id="ivorius"><a href="#">网页设计者：李四工作室</a></p>
    </div><!--end footer-->
```

上面代码中，footer层包含了一个段落信息，其中段落的id是ivorius。

在CSS文件中，用于修饰上面HTML标记的样式代码如下：

```
    #footer {
     background: transparent url('images/footer.png') no-repeat scroll left bottom;
    margin-top: 5px;
    padding-top: 2px;
    height: 33px;
    }
    #footer p { text-align: center; }
```

```
    #footer a { color: rgb(255, 255, 255); }
    #footer a:hover { color: rgb(223, 58, 0); }
    p#ivorius {
    float: right;
    margin-right: 13px;
     font-size: 0.75em;
    }
    p#ivorius a { color: rgb(80, 210, 122); }
```

上面代码中，#footer选择器定义了页脚背景图片、内外边距的顶部距离和高度。其他选择器定义了页脚部分文本信息的对齐方式、超级链接样式等。



19.3 整体调整

在前面的各个小节中，完成了首页中不同部分的制作，其整个页面基本上都已经成形。在制作完成后，需要根据页面实际效果作一些细节上的调整，从而更加完善页面整体效果。例如各块之间的padding和margin值是否与页面整体协调，各个子块之间是否协调统一等。页面效果调整前，在IE9.0中，浏览效果如图19-7所示。



图19-7 页面调整前效果

从图19-7中，可以发现页面段落没有缩进，页面右侧列表选项之间距离太小等。这时可以利用CSS属性进行调整，其代码如下：

```
    p { margin: 0.4em 0.5em; font-size: 0.85em;text-indent:2em; }
```

```
a { color: rgb(25, 126, 241); text-decoration: underline; }
a:hover { color: rgb(223, 58, 0); text-decoration: none; }
a img { border: medium none ; }
ul, ol { margin: 0.5em 2.5em; }
h2 { margin: 0.6em 0pt 0.4em 0.4em; }
h3, h4, h5 { margin: 1em 0pt 0.4em 0.4em; }
* { margin: 0pt; padding: 0pt; }
body { background: rgb(61, 62, 63) url('images/body.gif') repeat;
color: white; font-size: 1em; font-family: "Trebuchet MS",Tahoma,"Geneva
CE",lucida; }
```

上面代码中，全局选择器*设置了内外边距距离，body选择器设置了背景颜色、图片、字体大小，字体颜色和字形等。其他选择器分别设置了段落、超级链接、标题和列表等样式信息。页面调整后，在IE9.0中的浏览效果如图19-1所示。



19.4 专家解惑

1．Firefox如何设置div层居中呢？

有两种方式可以设置div层居中。一是div样式属性设置margin-left和margin-right的值为auto，代码如下：

```
margin: auto(主要是 margin-left,margin-right)
```

但此时IE浏览器不适用。二是body设置text-align时，div样式属性需要进行设置，方可居中。

2．如何设置div垂直居中呢？

首先使用代码设置

```
vertical-align:middle;
```

然后将行距增加到和整个div一样高，例如：

```
line-height:200px;
```

最后插入文字，就可以实现div层垂直居中了。但缺点是要控制内容不换行。



第 20 章

似水年华——女性个人网站

本章引言

随着网络发展，女性网站如雨后春笋一般，逐渐成为网络世界中一道光彩秀丽的风景线。这些优秀的女性网站强烈地吸引着越来越多的女性网民投入到互联网络的海洋中去。本章实例将以女性网站为题材，介绍DIV和CSS布局。







20.1 构思布局

本实例是一个介绍女性生活的个人网站，主要体现女性生活的方方面面。主体色彩采用黑色，强调女性对逝去日子的无奈和生活的神秘色彩。实例完成后，在IE9.0中，浏览效果如图20-1所示。



图20-1 女性网站首页

20.1.1 设计分析

女性个人网站是女性健康和情感的门户，内容一般包括女性健康、时尚、女性交友、女性情感、美容保健等。主要为女性提供前沿服饰、休闲、娱乐、健康等生活咨询。本实例重点是展现女性自信和生活的一面。

本网站是女性个人网站，其主要显示区域用于显示本人日志，在本区域可以编辑和管理日志文章，同时浏览者可以进行评论。页面左侧的导航链接，对女性生活的各个方面进行链接，方便浏览者快速进入感兴趣的话题。

20.1.2 排版架构

从图20-1页面效果可以看出，页面总体结构是上中下结构，即页头部分、页面主体和页脚部分，并且页面主体部分又分为左右版式，左侧显示菜单导航，右侧显示日志文章。总的来说，网站首页是一个混合的嵌套结构。其页面效果如图20-2所示。

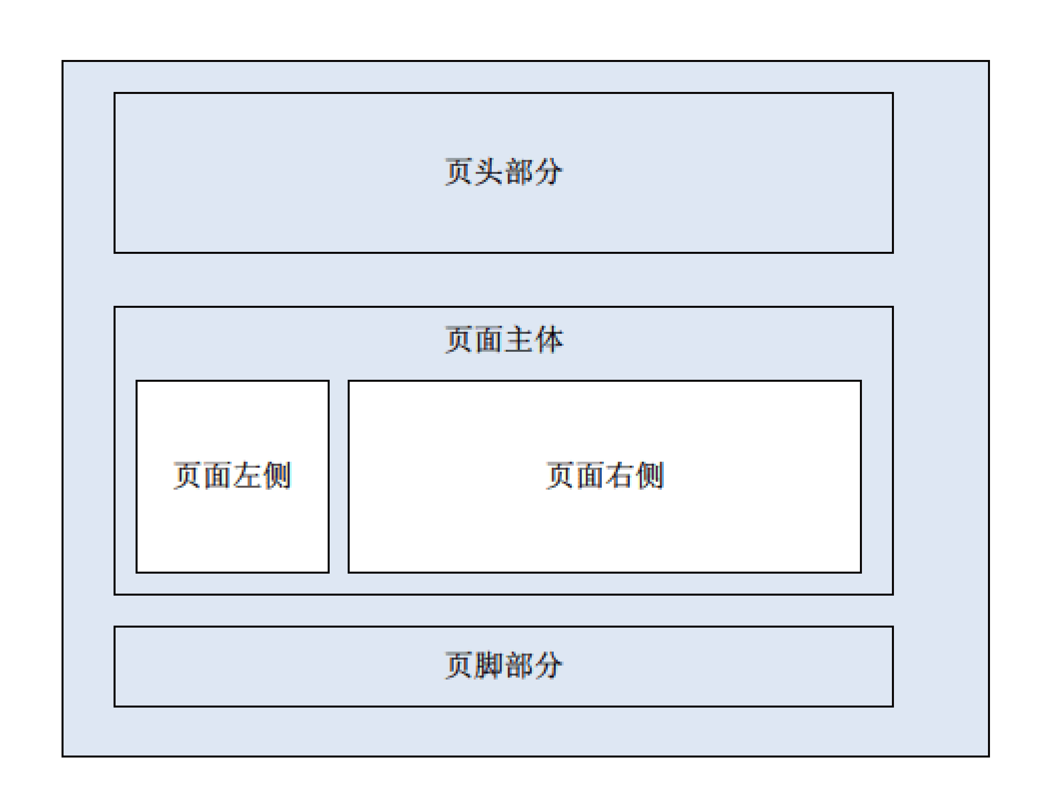


图20-2 页面总体框架

根据上面提供的页面框架，在HTML页面，可以使用div层将上面框架实现，其代码如下：

```
<div id="header">
</div>
<div id="page">
  <!-- start content -->
  <div id="content">
  </div>  <!-- end content -->
<div id="sidebar">  <!-- start sidebar -->
  </div>  <!-- end sidebar -->
</div><!-- end page -->
<div id="footer">
</div>
```

上面代码中，ID名称为header的层是页头部分，此处可以显示Logo和网站名称；ID名称为page的页面，是页面主体内容，包括两个不同的层，即content层和sidebar层。其中content层用于显示页面主体内容，即日志文章；sidebar层用于显示页面导航链接。footer层是页脚部分，用于显示地址和作者信息等。需要注意的是，此实例没有采用页面整体布局容器。

在CSS文件中，对应page层的修饰样式如下：

```
#page {
  width: 760px;
  margin: 0 auto;
}
body {
  margin: 0;
  padding: 0;
  background: #000000;
  font-size: 13px;
  color: #8C8C8C;
}
```

上面代码中，#page选择器定义了page层的显示宽度，即页面主体内容宽度和外边距距离等。标记选择器body中定义了内外边距距离为0、背景色为黑色、字体大小为13px以及字体颜色等。



20.2 模块分割

确定页面整体架构和各个子块之后，就可以动手对网页各个部分进行设计了。本实例对于整体样式修饰不是太复杂。本节主要完成各个子块的样式修饰。

20.2.1 网站名称

为了突出网站名称和色彩，网站导航链接没有放到页头部分，而是放到了页面主体中。本实例中页头部分包括两个部分，一个是网站名称；另一个是页面图片。图片和文字组合在一起，更加突出网站主体。在IE9.0中，浏览效果如图20-3所示。



图20-3 页头部分

根据上面效果，可以在HTML文件中创建并实现，其代码如下：

```
<div id="header">
  <h1>女性回忆</h1>
  <p>为爱而狂</p>
</div>
```

上面代码中，创建了一个层header，并且层里面包含了一个标题和一个段落。

在CSS文件中，对于上面标记的样式修饰代码如下：

```
#header {
  width: 760px;
  height: 280px;
  margin: 0 auto;
  background: url(images/img01.jpg) no-repeat right top;
}

#header h1, #header p {
  margin: 0;
  color: #EB42A3;
}

#header h1 {
  padding-top: 100px;
}
```

上面代码中，#header选择器定义了页头的宽度、高度、内边距和背景图等。其他选择器定义了页头部分标题和段落的样式，例如字体颜色和内外边距等。

20.2.2 日志文章

在页面主体中，右侧版式可以说是页面的主要部分，用于显示日志文章。并且每篇日志文章都可以进行评论、管理和编辑等。块的整体依然采用右浮动和固定宽度的版式。在IE9.0中，浏览效果如图20-4所示。



图20-4　页面右侧内容

根据图20-4所示的页面效果，在HTML文件中创建并实现，其代码如下：

```
    <div id="content">
        <div class="post">
          <h1 class="title">伊甸园日志</h1>
          <p class="byline"><small>2011年8月25发表 <a href="#">管理员</a> | <a
href="#">编辑</a></small></p>
          <div class="entry">
            <p><strong>喜欢回忆了么？</strong>花点时间去吧！去体验它，去找回它吧！<a
href="#">回忆</a> ，让我寻回真实．朋友。</p>
          </div>
          <p class="meta"><a href="#" class="more">更多</a>    
<a href="#" class="comments">评论 (33)</a></p>
        </div>
        <div class="post">
          <h2 class="title">花季</h2>
          <p class="byline"><small>2011年8月25日发表 <a href="#">管理员</a> | <a
href="#">编辑</a></small></p>
          <div class="entry">
          <p>时光飞逝总是让我们容易忽视很多与青春有关的日子……</p>
          <p>是不是总会在不惑之年才开始回忆过去的往事…… </p>
          </div>
          <p class="meta"><a href="#" class="more">更多</a>    
<a href="#" class="comments">评论 (33)</a></p>
        </div>
```

```
    </div>
```

上面代码中，创建了一个层content，用于包含页面中的日志文章信息。在content层中，又创建了两个匿名div，分别包含不同的日志文章。并且每个匿名div层中，都包含标题、段落和超级链接等信息。

在CSS文件中，用于修饰上面HTML标记的代码如下：

```
#content {
  float: right;
  width: 460px;
}
.post {
  padding: 0 0 20px 0;
}
.flowers {
  background-image: url( 'images/img03.jpg' )
}
.title {
  margin: 0;
  border-bottom: 2px solid #0F0F0F;
}
.byline {
  margin: 0;
  color: #646464;
}
.meta {
  text-align: right;
  color: #646464;
}
.meta .more {
  padding-left: 20px;
  background: url(images/img03.gif) no-repeat left center;
}
.meta .comments {
  padding-left: 20px;
  background: url(images/img04.gif) no-repeat left center;
}
```

上面代码中，#content选择器定义了层宽度为460像素，并且右浮动显示块。类选择器title定义了外边距和下边框显示样式，类选择器meta定义了对齐方式和字体颜色，类选择器more定义了左侧内边距距离和背景图片等信息。其他选择器多是定义了背景图片，这里就不再介绍了。

20.2.3 页面导航

女性是感性的，不习惯在页面中寻找超级链接，故本实例将所有页面导航链接放到一起，方便查找和发现。同样的，需要设置该块固定宽度和左浮动。在IE9.0中，浏览效果如图20-5所示。



图20-5 页面导航

根据图20-5所示的页面效果，在HTML文件中创建并实现，其代码如下：

```
<div id="sidebar">
    <ul>
      <li>
        <h2>女性生活</h2>
        <ul>
          <li><a href="#">美容</a></li>
          <li><a href="#">服饰</a></li>
          <li><a href="#">保健</a></li>
          <li><a href="#">娱乐</a></li>
          <li><a href="#">亲子</a></li>
          <li><a href="#">女包</a></li>
        </ul>
      </li>
      <li>
        <h2>心理情感</h2>
        <ul>
          <li>单身交友 (23)</li>
          <li><a href="#">恋爱部落</a> (31)</li>
          <li><a href="#">男婚女嫁</a> (31)</li>
          <li><a href="#">家庭婚姻</a> (30)</li>
          <li><a href="#">边缘情感</a> (31)</li>
          <li><a href="#">情感测试</a> (30)</li>
          <li><a href="#">明星情感</a> (31)</li>
          <li><a href="#">先锋女性</a> (28)</li>
          <li><a href="#">书生活</a> (31)</li>
        </ul>
      </li>
    </ul>
  </div>
```

上面代码中，创建了一个sidebar层，用于包含超级链接。在该层中，创建了标题和无序

列表，其中标题作为列表名称，无序列表作为导航菜单。

在CSS文件中，对于上面HTML标记进行修饰，其CSS代码如下：

```
#sidebar {
  float: left;
  width: 240px;
}
#sidebar ul {
  margin: 0;
  padding: 0;
  list-style: none;
}
#sidebar li ul {
  padding: 15px 0;
}
#sidebar li li {
  padding-left: 20px;
  border-bottom: 1px dotted #0F0F0F;
  background: url(images/img02.gif) no-repeat 5px 50%;
}
#sidebar h2 {
  margin: 0;
  padding: 20px 0 0 0;
  border-bottom: 2px solid #0F0F0F;
}
#sidebar a {
  text-decoration: none;
}
```

上面代码中，#sidebar选择器定义了层固定宽度为240px，页面左浮动。嵌套选择器#sidebar ul中定义了无序列表显示样式，例如内外边距和无特殊符号显示。其中选择器分别定义了列表选项、标题和超级链接的显示样式，例如背景色、边框样式和下划线等。

20.2.4 作者信息

本实例页脚信息比较简单，只是一个作者信息。在IE9.0中浏览效果如图20-6所示。

图20-6 页脚信息

在HTML文件中，创建并实现上述效果，其代码如下：

```
<div id="footer">
  <p>设计者 李四工作室</p>
</div>
```

上面代码中，创建了一个footer层，并且层中包含了一个段落。

在CSS文件中，对footer层的修饰代码如下：

```
#footer {
  clear: both;
  padding: 30px 0;
  background: url(images/img07.gif) repeat-x;
  text-align: center;
  font-size: smaller;
}
```

上面代码中，#footer选择器首先使用clear属性消除上面层使用float带来的影响，然后使用CSS属性定义了内边距、背景图片、对齐方式和字体大小等。



20.3 整体调整

完成对各个子块的排版后，网站基本上就能够实现了。最后还必须对整体样式进行调整，例如细节上的调整等。页面调整前，在IE9.0中浏览效果如图20-7所示。

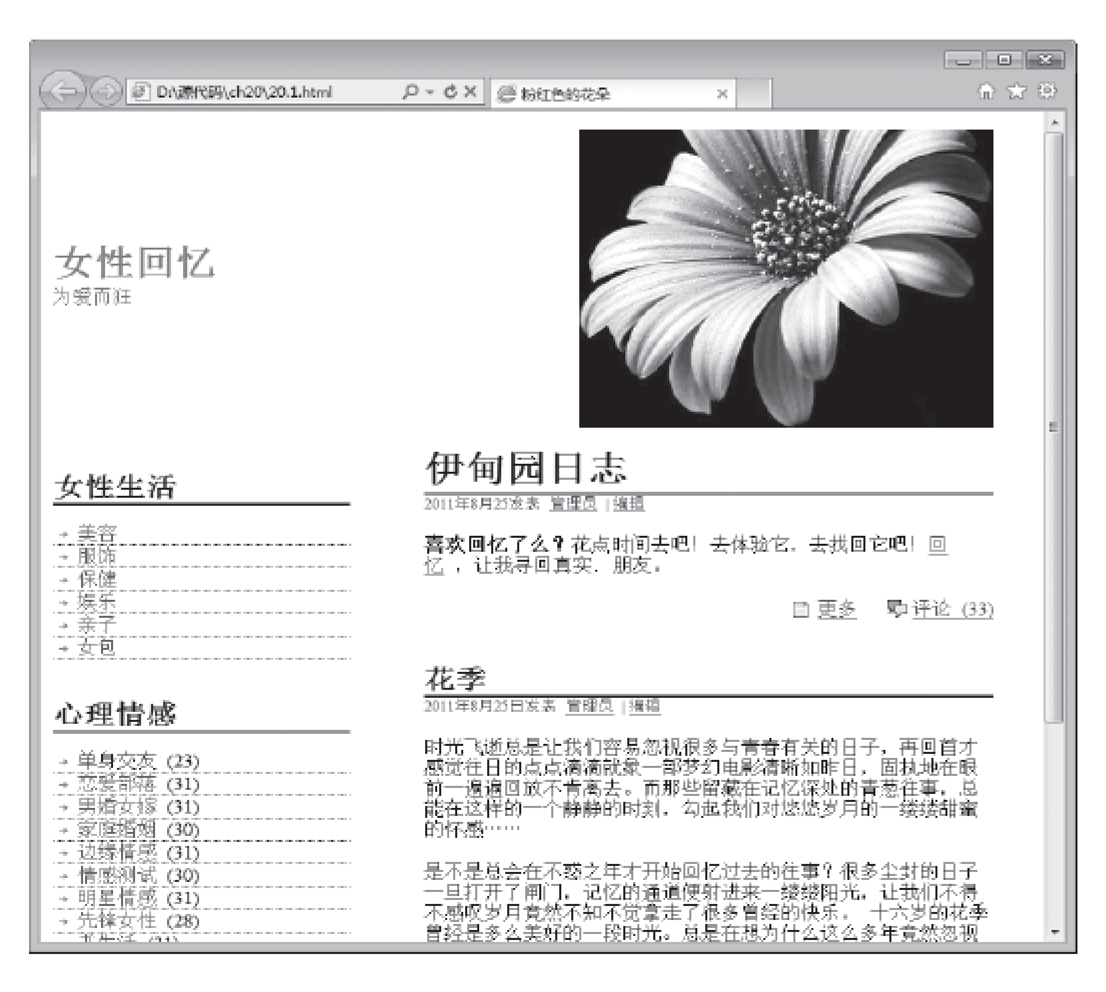


图20-7 调整前样式

CSS调整代码如下：

```
body, th, td, input, textarea, select, option {
  font-family: "宋体", "Times New Roman", Times, serif;
}
h1, h2, h3 {
  font-weight: normal;
  color: #CBA61A;
}
h1 {
  letter-spacing: -2px;
```

```
      font-size: 3em;
    }
    h2 {
      letter-spacing: -1px;
      font-size: 2em;
    }
    h3 {
      font-size: 1em;
    }
    p, ul, ol {
      line-height: 200%;
    }
    a {
      color: #3AA0B9;
    }
    a:hover {
      text-decoration: none;
      color: #EB42A3;
    }
```

上面代码中，定义了不同标记的显示样式，其标记包含标题h1、h2、h3、段落和超级链接等。样式大多为字体大小、字体颜色、背景色、字体和行高等。

页面调整后，可以发现字体颜色、字体大小和行高发生变化。在IE9.0中浏览效果如图20-1所示。

 20.4 专家解惑

1．在Firefox浏览器下，多层嵌套时内层设置了浮动，外层设置背景时，为什么背景不显示呢？

这主要是内层设置浮动后，外层高度在Firefox下变为0，所以应该在外层与内层间再嵌一层，设置浮动和宽度，然后再给这个层设置背景。

2．在IE浏览器中，如何解决双边距问题？

浮动元素的外边距会加倍，但与第一个浮动元素相邻的其他浮动元素外边距不会加倍。其解决方法：对此浮动元素增加样式display:inline。

3．元素定义外边距时，应注意哪些问题？

在对元素使用绝对定位时，如果需要定义元素外边距，应注意在IE6浏览器中，外边距不会视为元素的一部分，因此在对此元素使用绝对定位时外边距无效。但在Firefox浏览器中，外边距会视为元素的一部分，因此在对此元素使用绝对定位时，外边距有效（例如，margin_top会和top相加）。

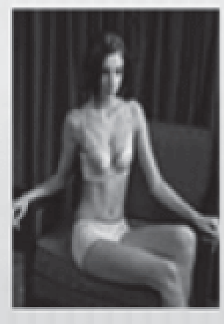


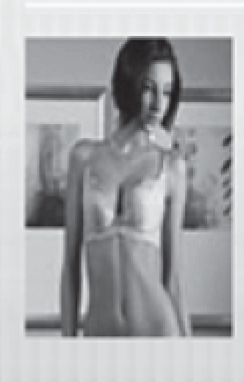
第 21 章

公司宣传有秘籍——时尚内衣网站

本章引言

作为公司的门户网站，其中主要职能就是宣传公司形象，展示产品。例如内衣公司，就需要宣传内衣产品。本章以内衣时尚网站为例，进一步介绍CSS构建网站的方法。









21.1 构思布局

本实例网站主要以介绍公司产品为主，采用图文混合的方法，并结合内衣主题，主色调采用浅色，容易被人接受。在Firefox 5.0中，浏览效果如图21-1所示。



图21-1 内衣时尚网站

21.1.1 设计分析

网站首页主要采用图文结合方式展现产品，其中图片用于显示新产品样式，文字作为辅助介绍。结合内衣行业内衣款式不断变化的情况，网站在页面主体的页面导航部分，用于新产品的展示。

页面主体部分分为品牌形象、产品展示和内衣欣赏等栏目，各个栏目都配有图片，浅蓝色背景，让页面在浅色下不失高雅、整洁和大方。页面导航链接在页头部分显示，作为链接其他页面的依据。对于公司信息，可以在页脚部分进行链接。

21.1.2 排版架构

网站首页的排版，包括Logo图片、导航菜单、左侧版式的新产品展示、右侧版式的品牌形象、内衣欣赏和产品展示等。其框架如图21-2所示。

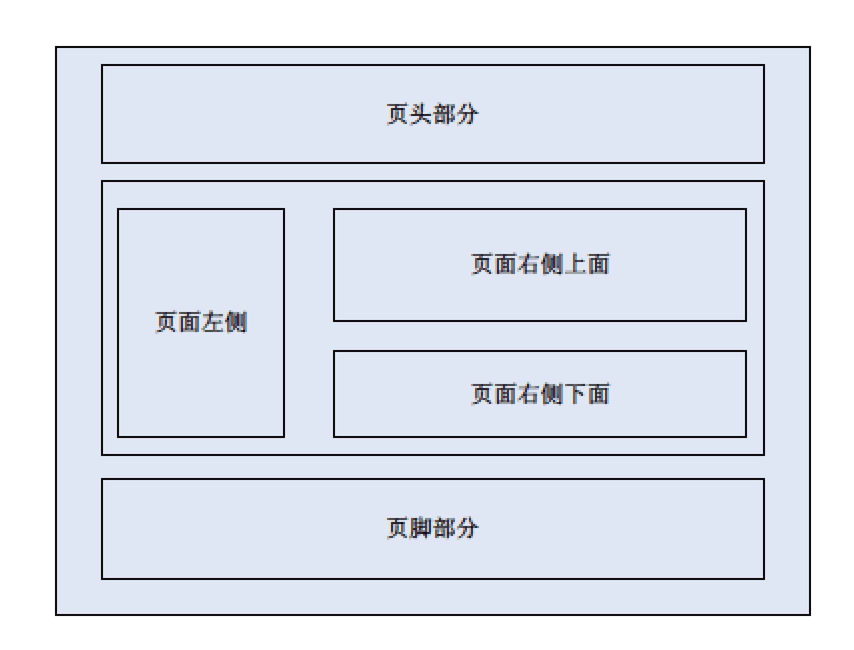


图21-2 页面框架

从图21-2所示框架图可以看出，页面总体是上中下结构，页面主体采用左右版式，但页面主体右侧又采用上下版式，相对于前面的实例，其排版有点复杂。结合框架图，在HTML文件中，使用div实现上面的架构，代码如下：

```
<div  class="main">                                          /*页面布局容器*/
  <div class="head">                                         /*页头部分*/
  <div class="menu">                                         /*页头部分*/
  </div>
  </div>
<div class="middmain">
    <div class="middmaintop"></div>
    <div class="leftmain">                                   /*页面主体左侧部分*/
    </div>
    <div class="rightmain">>                                 /*页面主体右侧部分*/
    <div class=" rightmaintopleft ">                         /*页面主体右侧上面*/
    </div>
    <div class="rightmaintopleft" style="float:right">       /*页面主体右侧上面*/
</div>
    <div class="litle" />内衣欣赏</div>                      /*页面主体右侧下面*/
    </div>
  </div>
</div>
  <div class="buttom"></div>                                 /*页脚部分*/
</div>
</body>
</html>
```

上面代码中，各个div层对应不同的框架区域，这里没有采用ID作为div层标识符，可以使用类选择器作为其标识符。main层用作布局容器，用于放置所有的HTML元素；head层和menu层共同组成页头部分，head层用于显示Logo图片，menu层显示导航菜单。middmain层是页面主体容器，用于存放页面主体的各个元素。leftmain层是页面主体左侧部分，用于新产品展示；rightmaintopleft层用于显示页面主体右侧上面的内容，这里面有两个层使用这样的样式，这两个层水平左右版式排列。title层用于显示页面主体右侧下面的内容，即内衣欣赏栏目。buttom层是页脚部分。

在CSS文件中，对于上面的布局容器main和middmaintop等，其样式代码如下：

```
.main{
```

```
        background:#DEF1F8;
        width:775px;
        margin:0 auto;
        border:1px #000 solid;
        overflow:hidden;}
    .middmain{
        float:left;
        background:url(img/maintitle2.jpg);
        width:775px;
                    }
    .middmaintop{
        width:775px;
        height:20px;
        float:left;}
```

上面代码中，类选择器main定义了层宽度为775px、边框样式、外边距和背景色。类选择器middmain定义了页面宽度、背景图和左浮动。类选择器middmaintop定义了宽度、高度和左浮动。



21.2 模块分割

在确定页面的各个块与子块后，就可以对网页的各个部分着手设计了。实例中块的整体布局样式这里就不再介绍了，前面小节已经介绍过。

21.2.1 页面导航和Logo图片

此处的页面导航菜单，采用了项目列表实现。将<ul>标记和<li>标记进行相应的设置，使得菜单能够显示在同一行上面。页面Logo非常简单，只是在图片上显示了网站名称。在Firefox 5.0中浏览效果如图21-3所示。



图21-3 页头部分

当鼠标放到导航菜单的超级链接上，其字体颜色会发生变化。在HTML文件中，实现上面效果的HTML代码如下：

```
      <div class="head">
      <div class="menu">
      <ul  >
       <li style="width:80px;"></li>
      <li ><a href="#">首页</a> </li>
      <li ><a  href="#">魔姿简介</a> </li>
      <li ><a  href="#">品牌形象</a></li>
      <li ><a href="#">产品展示</a> </li>
      <li ><a href="#">招商加盟</a> </li>
      <li ><a href="#">客户服务</a> </li>
      </ul>
      </div>
      </div>
```

上面代码中，定义了两个层head和menu，其中head层包含menu层。menu层中创建了一个无序列表，列表宽度使用CSS代码定义为80px。

在CSS文件中，对于上面标记的样式修饰代码如下：

```
    .head{
       overflow:hidden;
       width:775px;
       background:url(img/21.jpg);
       height:336px;
       float:left;}
    .menu{
       height:56px;
       float:left;
       padding-left:240px;
    }
    .menu li{
       padding-top:20px;
       font-size:14px;
       color:#FFF;
       font-weight:bold;
       width:75px;
       float:left;
                   }
    .menu ul li a{
                    color:#ddeecc;
                   display:block;
                  border:1px solid #AE9956;
                  background-color:#AE9956;
     }
    .menu a,.menu a:vistied,.menu a:link{
       text-decoration:none; color:#fff;}
    .menu a:hover{
       color:#F0C; text-decoration:underline;
    }
```

上面代码中，类选择器head定义了层宽度、高度、左浮动和背景图片；类选择器menu定义了层高度、左浮动和左侧内边距。嵌套选择器menu li定义了列表选项字体大小、宽度、

字体颜色、上侧内边距和左浮动。其他选择器定义了列表选项的超级链接样式，例如鼠标悬浮、激活状态的样式。

21.2.2 新产品展示

为了方便浏览者快速把握新产品动态，在首页专门开辟了一个新产品专栏，位于页面主体中的左侧部分。在Firefox 5.0中浏览效果如图21-4所示。

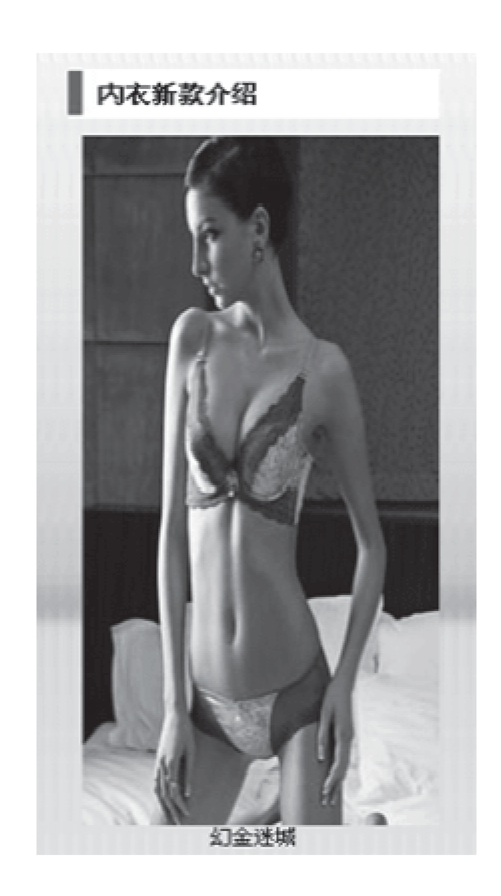


图21-4 新产品展示

根据图21-4所示的页面效果，在HTML文件中实现。其代码如下：

```
<div class="leftmain">
      <div class="leftmaindiv">
        <div class="litle"><img src="img/ltitlekuai.jpg" border="0"
/>内衣新款介绍</div>
    <div class="leftmainpic">
          <a href="#"><img src="img/35.jpg" border="0" /></a><br />
               <p style="text-align:center;"> 幻金迷城 </p>
        </div>
      </div>
    </div>
```

上面代码中，定义了leftmain层和leftmiandiv层作为页面主体左侧布局容器；在leftmaindiv层中包含了title层和leftmainpic层，其中title层用于显示左侧栏目标题；leftmainpic层用于显示图片信息和文本信息。

在CSS文件中，对于上面HTML标记的样式代码如下：

```
.leftmain{
   width:220px;
   padding-right:30px;
   padding-left:20px;
   float:left;
             }
.leftmaindiv{
   background:#f9f8cb;
```

```
        width:208px;
        border:1px #FFF solid;
        }
    .leftmainpic{
        font-size:12px;
        width:208px;
        float:left;
        padding:10px 0;}
    .leftmainpic a img{
                    float:left;
                    padding-left:8px;
    }
    .leftmainpic p{
                   text-align:center;
                   font-size:12px;
    }
    .litle{
        font-size:14px;
        font-weight:bold;
        width:208px;
        float:left;
        height:28px;
        background:#FFF;
        line-height:28px;
    }
    .litle img{
                  float:left;
                  height:28px;
    }
```

上面代码中，类选择器leftmain中定义了层宽度、左右侧内边距和页面左浮动。类选择器leftmaindiv定义了背景色、宽度和边框样式；选择器leftmainpic定义了字体大小、宽度、浮动和内边距等。类选择器litle定义了字体大小、字体样式、宽度、浮动、高度、背景色和行高等。其他选择器在上面选择器的基础上，对其包含元素进行样式设计，例如对齐方式、高度等。

21.2.3 品牌形象和产品展示栏目

在页面主体右侧部分上面，品牌形象和产品展示栏目处于主要地位。二者采用左右版式结构，水平排列在版面上。块的整体实现采用了浮动和固定宽度的版式。在Firefox 5.0中浏览效果如图21-5所示。



图21-5 页面主体右侧

在HTML文件中实现图21-5所示的页面效果，其代码如下：

```
        <div class="rightmain">
    <div class="rightmaintopleft">
        <div class="lefttitle">品牌形象</div>
        <img src="img/xian.jpg" border="0" style="width:230px; float:left;">
        <div class="leftmaintopleftcontent"><a href="#"><img src="img/5.jpg"
border="0" style="float:left; padding-top:10px;" ></a><font >
     魔姿旗下拥有的品牌及产品线包括：“魔姿女士”、“魔姿先生”、“魔姿儿童”、“魔姿美体内
衣”，“爱美丽”、“兰卡文”、“心爱”，以及个性化内衣和礼服定制工作室。</font>
     <div class="downcontent">  <a href="#">独享魔姿内衣品牌,穿出性感,穿出自信,穿出
健康,自主研发生产,平民的价格,一流的质量,一站式供货。  ！</a> </div>
     </div>
        </div>
        <div class="rightmaintopleft" style="float:right">
        <div class="lefttitle">产品展示</div>
        <img src="img/xian.jpg" border="0" style="width:230px; float:left;">
        <div class="leftmaintopleftcontent"><a href="#"><img src="img/6.jpg"
border="0" style="float:left; padding-top:10px;" ></a><font >
        本系列为11年得特别限量版，采用欧洲顶级的刺绣花边，经典的杯型、颜色设计，彰显精致与尊
贵。炫彩蟒纹流线造型配合暗纹印花，性感魅惑，打造专属于LACLOVER女人的高贵的野性风格。
       </font>
          <div class="downcontent">    <a href="#">致力于为女性提供健康舒适,时尚美
体的高性价比内衣商品和服务.努力让每一位东方女性更加美丽自信.</a> </div>
     </div>
        </div>
    </div>
```

上面代码中，创建了多个div层，其中比较重要的是使用类选择器rightmaintopleft修饰的层。这两个层分别是图21-5中的品牌形象和产品展示。lefttitle层主要存放栏目标题，leftmaintopleftcontent层用于包含文本介绍信息，downcontent层用于包含链接信息。rightmain层是页面主体右侧布局容器。

在CSS文件中，对于上面HTML标记，其样式代码如下：

```
    .rightmain{
       width:490px;
       float:left;
               }
    .rightmaintop{
       width:490px;
```

```
        float:left;
        padding-bottom:20px;
        }
    .rightmaintopleft{
        width:230px;
        float:left;
        height:270px;}
    .lefttitle{
        font-size:14px;
        width:220px;
        float:left;
        height:28px;
        line-height:28px;
        padding-left:10px;
        font-weight:bold;

    }
    .leftmaintopleftcontent{
        width:230px;
        float:left;
    }
    .leftmaintopleftcontent font{
    font-size:12px;
     float:left;
    line-height:20px;
    padding-top:10px;
    width:132px;
    padding-left:10px;
    }
    .downcontent{
        width:228px;
        float:left;
        font-size:12px;
        line-height:20px;}
    .downcontent a,.downcontent a:vistied,.downcontent a:link{
        text-decoration:none; color:#000;}
    .downcontent a:hover{
        color:#666; text-decoration:underline;}
```

上面代码中，类选择器rightmain定义层宽度和左浮动；层rightmaintop定义页面宽度、左浮动和底部内边距；层rightmaintopleft定义页面宽度、高度和左浮动；层lefttitle定义字体大小、宽度、左浮动、高度、行高、字体样式和左侧内边距。其他选择器分别定义了段落、字体的样式等。

21.2.4 内衣欣赏

在页面主体右侧部分下面，是内衣欣赏栏目。内衣欣赏栏目实际上是一个图片栏目，将比较有代表性的图片在该栏目展示，用于吸引人的目光。图片展示同样采用无序列表，即将图片作为列表选项。在Firefox 5.0中浏览效果如图21-6所示。

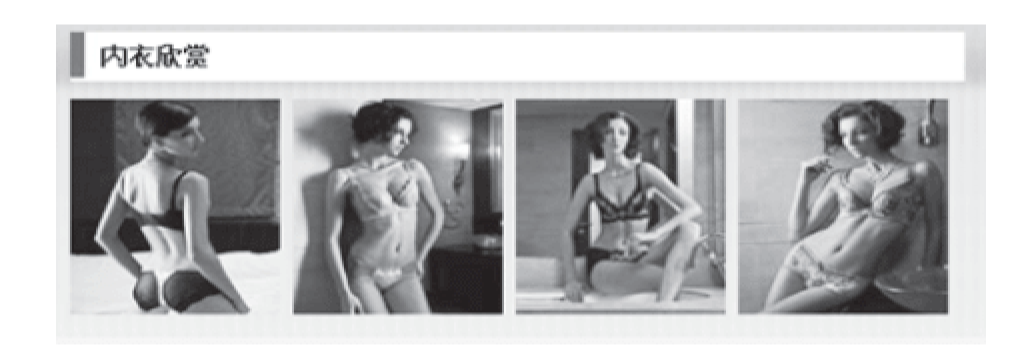


图21-6 内衣欣赏栏目

在HTML文件中，实现图21-6所示的页面效果。其HTML代码如下所示。

```
        <div class="litle" style="width:490px;"><img src="img/ltitlekuai.
    jpg" border="0" height="28" style="float:left;" />内衣欣赏</div>
        <div class="pic">
        <ul><li><div style="width:116px"><a href="#"><img src="img/7.jpg"
border="0"></a></div></li><li><div style="width:116px"><a href="#"><img
src="img/8.jpg" border="0"></a></div></li><li><div style="width:116px"><a
href="#"><img src="img/9.jpg" border="0"></a></div></li><li><div
style="width:116px"><a href="#"><img src="img/10.jpg" border="0"></a></div></
li></ul>
        </div>
```

在上面代码中，创建了两个div层，litle层用于包含页面栏目标题。pic层用于显示产品图片，其图片在列表中显示，图片宽度为116px。

在CSS文件中，对上面标记进行修饰，其代码如下：

```
    .pic{
       width:490px;
       height:118px;
       float:left;
       padding-top:10px;
       }
    .pic ul{
       width:490px;
       list-style-position:outside;
    }
    .pic li{
       padding-right:6px;
       float:left;
       list-style-type:none;
       list-style-position:outside;
       }
```

上面代码中，类选择器pic定义了层宽度、高度、左浮动和顶部内边距；嵌套选择器pic ul定义了无序列表宽度和样式显示位置；嵌套选择器pic li定义了右侧内边距、左浮动、无符号显示等。

21.2.5 公司信息

本实例的公司信息显示在页脚部分，方便浏览者查看，当然其版权信息和作者信息也是在这个地方显示。在Firefox 5.0中浏览效果如图21-7所示。

关于我们 | 网站地图 | 诚聘英才 | 零售店址 | 网站联盟 | 网店代理
CopyRight 2011-2019@MzChina.All Rights Reserved

图21-7　公司信息

在HTML文件中，实现上述效果，其代码如下：

```
    <div class="button"><a href="#">关于我们</a> | <a href="#">网站地图</a>
| <a href="#">诚聘英才 </a> | <a href="#">零售店址</a> | <a href="#">网站联盟 </a>
| <a href="#">网店代理 </a>
  <br>CopyRight 2011-2019@MzChina.All Rights Reserved
  </div>
```

在上面代码中，创建了一个div层，用于包含页脚信息。层里面包含了多个超级链接，例如使用超级链接显示常见的公司信息等。

在CSS文件中，对于上面HTML标记进行修饰，其代码如下：

```
.buttom{
    float:left;
    font-size:12px;
    width:775px;
    padding-top:15px;
    text-align:center;
    line-height:20px;}
.buttom a,.buttom a:vistied,.buttom a:link{
    text-decoration:none; color:#000;}
.buttom a:hover{
    color:#666; text-decoration:underline;}
```

在上面代码中，类选择器buttom定义了页面左浮动、字体大小、宽度、顶部内边距、对齐方式和行高等。其他选择器定义了超级链接显示样式，例如访问、激活和悬浮样式。



21.3　整体调整

通过对上面各个子块的排版，整个网站到此时就基本完成了。最后必须对页面整体进行查看，细节上作小的调整。页面调整前后效果相差不大，这里就不再单独显示调整前的效果了。对页面调整的代码如下：

```
*{
    margin:0;
    padding:0;}
.body{
    font-size:12px;
    color:#000;
```

```
    font-family:"宋体";}
.ul ,li{
    list-style-type:none;}
```

上面代码中，*全局选择器定义内外边距距离都为0，body选择器定义了字体大小、字体颜色和字体等。

调整后页面效果，在Firefox 5.0中，浏览如图21-1所示。



21.4 专家解惑

1．Firefox和IE浏览器，如何处理负边界问题？

在IE浏览器中，对于超出父元素的部分会被父元素覆盖，而在Firefox浏览器中，对于超出父元素的部分会覆盖父元素，但前提是父元素有边框或内边距，不然负边距会显示在父元素上，使得父元素拥有负边距。在进行网页设计时，针对上面的情况可以对元素进行相对定位。

2．在定义子元素的上边距时，如果超出元素高度，怎么处理？

在IE浏览器中，子元素上边距显示正常，而在Firefox浏览器中，子元素上边距显示在父元素上方。其解决办法是在父元素中增加overflow:hidden语句或给父元素增加边框或内边距。

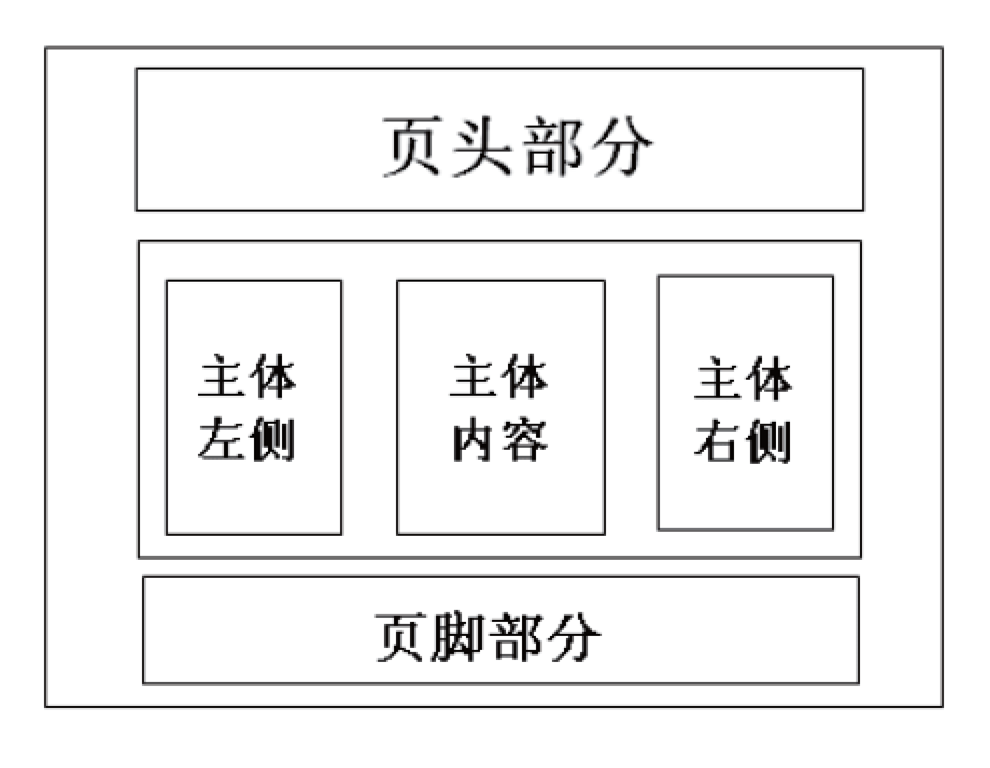


第 22 章

传承文化——文化宣传类网站

本章引言

在众多网站中，用于介绍诗歌、小说和文化的网站是其中的一种题材。这种网站一般都给人以整洁和大方的感觉，其内容主题比较单一。本章将以一个介绍文化的网站为例，进一步介绍DIV和CSS布局。







22.1 构思布局

本实例以文化宣传为中心，介绍人文中国、山水文化、文化天地、文化热点、旅游文化、美食文化和建筑文化等栏目。在IE9.0中，浏览效果如图22-1所示。



图22-1 文化宣传网站

22.1.1 设计分析

文化是一个群体，例如可以是国家、也可以是民族、企业和家庭等，在一定时期内形成的思想、理念、风俗和习惯，及由这个群体整体意识所衍生出来的一切活动。此类网站的作用就是宣传文化资料，用于教育和引导浏览者。

页面左右放置了多个超级链接，方便浏览者查看自己喜欢的内容，风格上则配合整体的页面设计，白色背景配合浅色边框，将文化的简洁性和传承性一一体现。

整个版面采用固定宽度且居中的版式，对于大显示器的用户，两边将显示浅灰色背景，将整个页面衬托出来，显得主次分明。

22.1.2 排版架构

整个页面框架采用上中下总体结构，页面主体内容版式采用左中右的版式，主要包括Logo、网站名称、页面导航、人文中国、站点搜索、文化天地和文化热点等。其框架如图22-2所示。

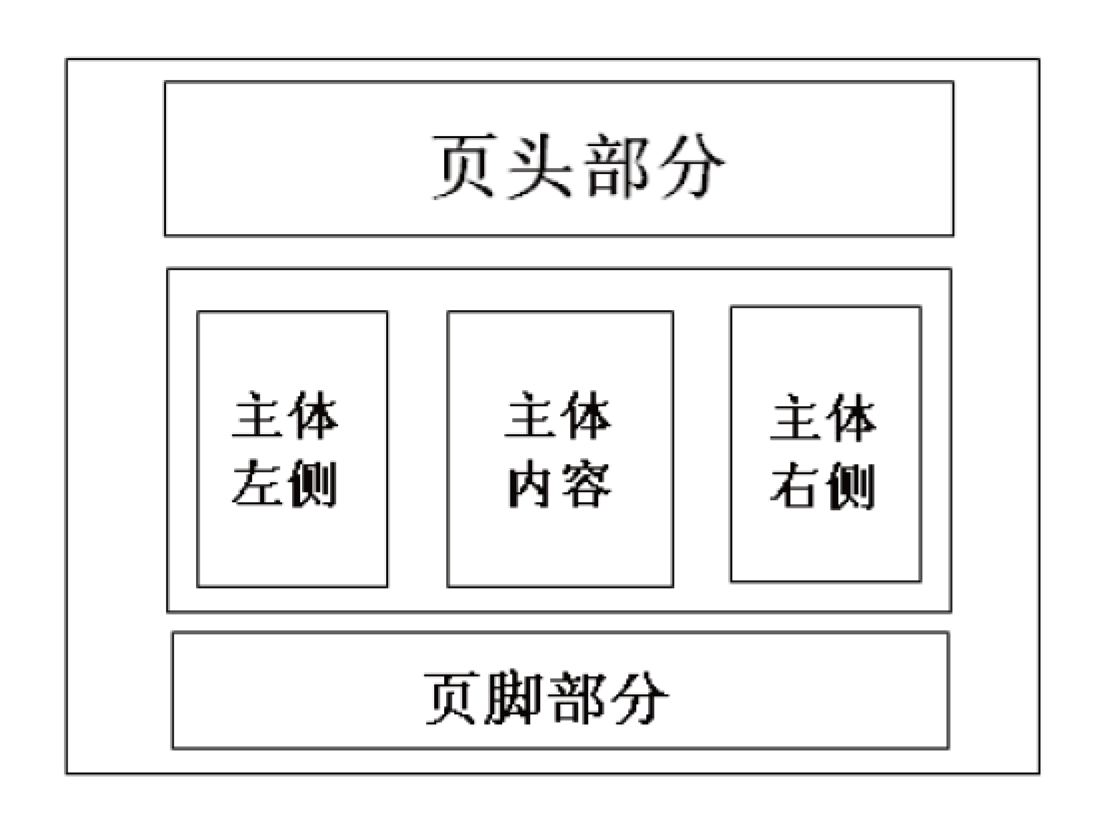


图22-2 页面框架图

从图22-2中可以看出，版式采用嵌套版式。在HTML文件中，使用div层实现上面各个子块的划分。其代码如下：

```
<div id="wrap">
<div id="header">              /*页头部分*/
</div>
<div id="leftside">            /*页面主体左侧*/
</div>
<div id="extras">              /*页面主体右侧*/
</div>
<div id="content">             /*页面主体中间部分*/
<div class="post">             /*页面主体中间部分上*/
</div>
<div class="post">             /*页面主体中间部分下*/
</div>
</div>
<div id="footer"></div>        /*页脚部分*/
</div>
</body>
</html>
```

上面代码中，层wrap是布局容器，用于存放页面中所有的HTML元素；层header是页头部分，存放网站名称和Logo；层leftside是页面主体左侧部分，用于存放页面导航菜单；层content是页面主体中间部分，用于存放文本信息；层extras是页面主体右侧部分，包括页面搜索和导航菜单等；层footer是页脚部分，存放作者信息等。

在CSS文件中，对于布局容器wrap的样式代码如下：

```
#wrap {
 margin: 0pt auto;
background-color: rgb(255, 255, 255);
color: rgb(48, 48, 48);
width: 760px;
}
```

上面代码中，定义了布局容器wrap中的样式，例如宽度、颜色、背景色和外边距等。



22.2 模块分割

整体框架确定好后，就可以对个子块进行设计了。各个子块完成后，还需要对整体样式进行调整。

22.2.1 页头部分

本实例的页头部分，包括两个部分，一个是网站名称，另一个是页头背景图片，相当于页面Logo。其效果如图22-3所示。



图22-3 页头部分

在HTML文件中，实现上面效果的代码如下：

```
<div id="header">
<h1><span lang="zh-cn">中国文化</span></h1>
</div>
<img id="frontphoto" src="images/5d.jpg" alt="" height="175" width="760" />
```

上面代码中，创建一个div层和图片img，其中div层中包含了一个标题。

在CSS文件中，对于上面标记的CSS代码如下：

```
#header { margin: 10px 0pt 0pt; }
#header h1{
margin: 0pt 0pt 10px;
background-color: inherit;
color: rgb(80, 80, 80);
float: left;
font-size: 2em;
letter-spacing: -1px; width: 350px;
}
#frontphoto { margin: 0pt 0pt 10px; }
```

上面代码中，选择器header定义了外边距距离；选择器header h1定义了标题显示样式，例如外边距、背景色、字体颜色、左浮动、字体大小和字符间距等。选择器frontphoto定义了图片显示位置，例如外边距等。

22.2.2 左侧部分

页面主体内容版式分为左中右版式，此小节首先介绍主体内容左侧部分。左侧部分用于显示导航菜单，例如链接不同类别的文化栏目。其显示采用了固定宽度和左浮动的方式。在IE9.0中，浏览效果如图22-4所示。



图22-4 左侧导航菜单

在HTML页面中，实现上面效果的代码如下：

```
<div id="leftside">
<h2 class="hide">Main menu:</h2>
<ul class="page">
<li class="page_item"><a href="#" title="Page 1">首页</a></li>
<li class="page_item"><a href="#" title="Page 1">民间文化</a></li>
<li class="page_item"><a href="#" title="Page 2">美食文化</a></li>
<li class="page_item"><a href="#" title="Page 2">旅游</a></li>
<li class="page_item"><a href="#" title="Page 2">宗教</a></li>
<li class="page_item"><a href="#" title="Page 2">艺术</a></li>
<li class="page_item"><a href="#" title="Page 2">建筑文化</a></li>
</ul>
</div>
```

上面代码中，创建了一个div层，层中包含了一个标题和一个无序列表，无序列表显示导航菜单。

在CSS文件中，对于上面标记进行修饰，其代码如下：

```
    #leftside { margin: 0pt 0pt 5px; clear: left; float: left; line-height:
1.4em; width: 140px; }
    #leftside p, #leftside ul, #leftside div.textwidget, #extras p, #extras
ul, #extras div.textwidget { margin: 0pt 0pt 18px; font-size: 0.9em;
    }
    #leftside li, #extras li { margin: 0pt 0pt 6px; line-height: 1.2em; list-
style-type: none; list-style-image: none; list-style-position: outside; }
    #leftside ul ul, #extras ul ul { margin: 6px 0pt 0pt 10px; }
    #leftside ul ul li, #extras ul ul li { margin: 0pt 0pt 4px; }
    #leftside ul ul li a, #extras ul ul li a { font-weight: normal; }
    #leftside ul.linklist, #extras ul.linklist { font-size: 1em; }
```

```
    #leftside ul.linklist ul, #extras ul.linklist ul { margin: 0pt 0pt 18px; }
    #leftside ul.linklist ul li a, #extras ul.linklist ul li a { font-weight:
bold; }
    #leftside h2, #leftside ul.linklist li h2, #leftside h3, #extras h2,
#extras ul.linklist li h2, #extras h3 { margin: 0pt 0pt 10px; background-
color: inherit; color: rgb(80, 80, 80); font-size: 1.2em; }
    #leftside label, #extras label { display: none; }
    .hide { display: none; }
```

上面代码中，选择器leftside定义了外边距、行高、宽度和左浮动，并使用clear属性消除上面属性float的影响。其他选择器分别定义了无序列表选项和超级链接的显示样式。

22.2.3 右侧部分

网页主体内容右侧部分，自上而下包含三个栏目，即站点搜索、文化天地和文化热点。其中站点搜索用于浏览者在站内搜索自己喜欢的文章，此处使用表单创建；下面两个栏目都是使用无序列表创建。在IE9.0中，浏览效果如图22-5所示。



图22-5 主体右侧部分

在HTML文件中，实现上面效果的代码如下：

```
    <div id="extras">
    <h2>站点搜索</h2>
    <form method="get" id="searchform" action="#">
    <div>
    <label for="s">查询</label>
    <input value="" name="s" id="s" size="14" type="text" />
    <input id="searchsubmit" value="Search" type="hidden" />
    </div>
    </form>
    <h2>文化天地</h2>
    <ul>
       <li><a href="#" title="News1">文化视界</a> <br /></li>
       <li><a href="#" title="News1">文化遗产</a> <br /></li>
       <li><a href="#" title="News1">文化要闻</a> <br /></li>
       <li><a href="#" title="News1">人文中国</a> <br /></li>
```

```
        <li><a href="#" title="News1">走遍中国</a> <br /></li>
                  <li><a href="#" title="News1">文化人物</a> <br /></li>
    </ul>
    <h2>文化热点</h2>
    <ul>
        <li><a href="#" title="易经热分析">易经热分析</a></li>
        <li><a href="#" title="儒家文化回归">儒家文化回归</a></li>
        <li><a href="#" title="再论中国隐士文化">再论中国隐士文化</a></li>
                   <li><a href="#" title="中国文化之根---祭祀文化">中国文化之根---
祭祀文化</a></li>
    </ul>
    </div>
```

在上面代码中，首先创建了一个层div，其名称为extras。里面包含了表单信息、无序列表和标题的元素。

在CSS文件中，对于上面标记的修饰代码如下：

```
    #extras {
    margin: 0pt 0pt 5px;
    float: right;
    line-height: 1.4em;
    width: 140px;
    }
    #s {
    border: 1px solid rgb(204, 204, 204);
    margin: 0pt 0pt 16px;
    padding: 4px;
    background-color: rgb(255, 255, 255);
    color: rgb(80, 80, 80);
    font-size: 0.9em;
    width: 130px; }
```

上面代码中，选择器#extras定义了层的宽度、行高、右浮动和外边距等。#s选择器定义了表单元素显示样式，例如边框样式、外边距、内边距、背景色、字体颜色、字体大小和宽度等。这里需要注意的是，对于右侧部分的无序列表样式代码，是和22.2.2节中的选择器leftside一起定义的，这里就不再重复介绍了。

22.2.4 中间部分

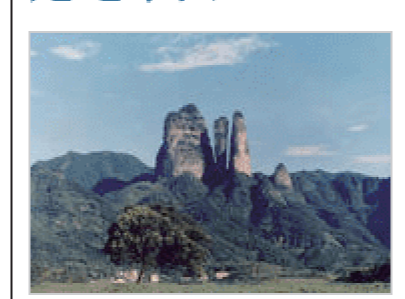
页面主体内容中间部分信息比较简单，只是显示两个栏目中的文章。自上而下可以分为人文中国和走遍中国两个栏目。在IE9.0中浏览效果如图22-6所示。

人文中国



斗茶是在品茶的基础上发展起来的。品茶也称品茗，由主人邀请三五知己，品评名茶。但是斗茶作为排列新茶优劣顺序的活动，则与平和的品茗不同，有着比技巧、斗输赢的特点。“斗”，正是在争斗中逞强获胜之意。曾经有私塾老师以“较筐箧之精，争鉴裁之别”来概括斗茶的涵义，因此这种比赛活动可谓非常富有挑战性和趣味性。

走遍中国



古江山城南江郎山通海。东海龙君海公主偕其地江氏三郎青梅竹马，情深谊笃。三序业儒，二郎捕鱼，大郎狩猎。龙君以海公主赐婚将军蟾蜍，俗谓癞蛤蟆者也，貌寝性贪。海公主不从，奔居江家。蟾蜍妒怒甚，率鳖精及虾兵蟹将、蜈蚣毒蛇无数，汹汹杀来。二郎举网捕虾蟹，掷石杀绝；三郎惟

图22-6 页面主体中间部分

在HTML文件中，实现上面效果的代码如下：

```
    <div id="content">
    <div class="post">
    <h2><a href="#" rel="bookmark" title="An image in a post">人文中国</
a></h2>
    <div class="contenttext">
    <p><img src="images/01.jpg" alt="an image" width="438" />
    斗茶是在品茶的基础上发展起来的。品茶也……
    </p>
    </div>
    <div class="post">
    <h2><a href="#" title="Blockquotes galore">走遍中国</a></h2>
    <div class="contenttext">
    <p><img src="images/江郎山.jpg" alt="an image" />古江山城南江郎山通海。东海龙君
海公主偕其地江氏三郎青梅竹马，……
    </p>
    <p><a href="#">更多选择…</a></p>
    </div>
    </div>
    </div>
    </div>
```

在上面代码中，层content是中间部分的布局容器，用于包含文本段落信息。该层中包含了标题、段落和图片等HTML元素。

在CSS样式文件中，用于修饰上面HTML标记的样式代码如下：

```
#content {
border-left: 1px solid rgb(224, 224, 224);
border-right: 1px solid rgb(224, 224, 224);
 margin: 0pt 150px 5px;
padding: 0pt 10px;
line-height: 1.6em;
}
#content h2 { margin: 0pt 0pt 10px; font-size: 1.6em; }
#content h3  { margin: 0pt 0pt 8px; font-size: 1.4em; }
#content img {
border: 1px solid rgb(208, 208, 208);
 margin: 3px 10px 3px 0pt;
float: left; }
#content ul,  #content ol, { margin: 0pt 0pt 16px 20px; }
#content li { padding: 0pt 0pt 0pt 5px; }
#content ul ul,  #content ol ol{ margin: 0pt 0pt 0pt 16px; }
.contenttext { overflow: hidden; }
* html .contenttext { overflow: visible; height: 1px; }
* html .contenttext p { overflow: hidden; width: 99%; }
.post { margin: 0pt 0pt 25px; }
```

上面代码中，#content选择器定义了左右边框显示样式、内外边距和行高。类选择器post定义了外边距，选择器contenttext定义了隐藏属性。其他选择器分别定义了标题、图片、列表选项等显示样式。

22.2.5 页脚部分

页脚部分通常用来显示版权信息、联系方式和作者信息等。通常只要风格与整体页面统一协调即可。本实例的页脚部分非常简单，在IE9.0中，浏览效果如图22-7所示。

版权所有：李四工作室

图22-7　页脚部分

在HTML文件中，实现上面效果的代码如下：

```
<div id="footer"><a href="#">版权所有：李四工作室</a></div>
```

上面代码中，创建了一个div层，层里面包含了一个超级链接。

在CSS文件中，修饰上面标记的样式代码如下：

```
#footer {
border-top: 2px solid rgb(224, 224, 224);
margin: 0pt auto;
padding: 10px 0pt;
background-color: rgb(255, 255, 255);
clear: both;
color: rgb(128, 128, 128);
font-size: 0.9em;
```

```
    font-weight: normal;
     line-height: 1.3em;
     text-align: center;
    width: 760px;
    }
    #footer a {
     background-color: inherit;
    color: rgb(128, 128, 128);
    font-weight: normal;
    text-decoration: none; }
    #footer a:hover {
    background-color: inherit;
    color: rgb(80, 80, 80);
    text-decoration: underline;
    }
```

上面代码中，选择器#footer定义了边框样式、内外边距、背景色、字体颜色、字体大小、字体样式、对齐方式和高度，并用clear属性消除上面使用float的效果。其他选择器分别定义了超级链接的显示样式，例如鼠标悬浮等。



22.3 整体调整

各个子块完成之后，其页面基本布局差不多就可以完成了。但最后还需要对页面整体样式进行查看，细节上做小的调整，例如调整padding和margin的值等。调整前，在IE9.0中浏览效果如图22-8所示。



图22-8 调整前效果

在CSS文件中，添加调整代码如下：

```
    * { margin: 0pt; padding: 0pt; }
    body {
    background: rgb(244, 244, 244) url('images/bg.gif') repeat-y scroll
center top;
    color: rgb(48, 48, 48);
    font-family: Verdana,Tahoma,Arial,sans-serif;
     font-style: normal;
    font-variant: normal;
    font-weight: normal;
    font-size: 76%;
     line-height: normal;
    font-size-adjust: none;
    font-stretch: normal;
     }
    a { background-color: inherit; color: rgb(40, 110, 160); font-weight:
bold; text-decoration: none; }
    a:hover { background-color: inherit; color: rgb(40, 110, 160); font-
weight: bold; text-decoration: underline; }
    a img { border: 0pt none ; }
```

在上面代码中，全局选择器*定义了内外边距，body选择器定义了背景色、背景图片、字体、字体颜色、字体大小、行高等。其他选择器分别定义了超级链接显示样式。

页面调整后，在IE9.0中，浏览效果如图22-1所示。可以发现字体大小、字体颜色和超级链接等发生了变化。



22.4 专家解惑

1．定义块的高度时，如果高度小于10px，则显示10px，如何解决？

在浏览器中，如果块元素最小高度为10px，当高度定义小于10px时仍为10px。其解决方法是，为此块元素添加样式overflow:hidden语句，或者让此块元素的字体大小等于此块元素的高度。

2．在浏览器中，若列表选项li为浮动，则列表后面的元素不能换行，如何解决？

其解决办法是，为这个ul无序列表定义合适的高度，或者给包含这个ul无序列表的父元素div层定义合适的高度。